

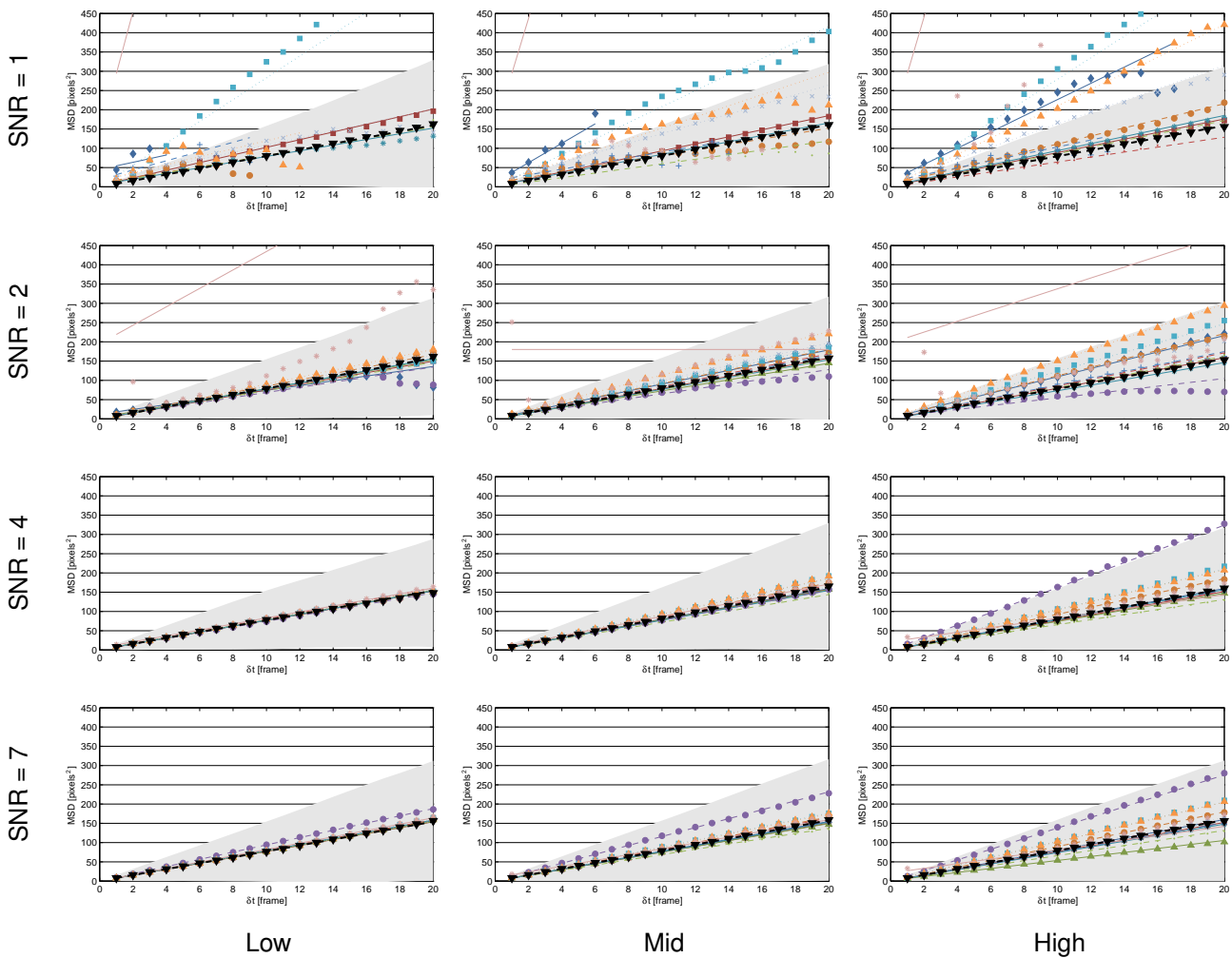
Objective comparison of particle tracking methods

Nicolas Chenouard, Ihor Smal, Fabrice de Chaumont, Martin Maška, Ivo F. Sbalzarini, Yuanhao Gong, Janick Cardinale, Craig Carthel, Stefano Coraluppi, Mark Winter, Andrew R. Cohen, William J. Godinez, Karl Rohr, Yannis Kalaidzidis, Liang Liang, James Duncan, Hongying Shen, Yingke Xu, Klas E. G. Magnusson, Joakim Jaldén, Helen M. Blau, Perrine Paul-Gilloteaux, Philippe Roudot, Charles Kervrann, François Waharte, Jean-Yves Tinevez, Spencer L. Shorte, Joost Willemsse, Katherine Celler, Gilles P. van Wezel, Han-Wei Dan, Yuh-Show Tsai, Carlos Ortiz de Solórzano, Jean-Christophe Olivo-Marin, Erik Meijering

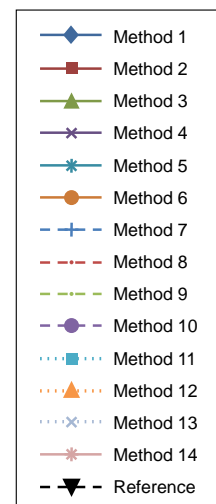
Supplementary Information

- Supplementary Figure 1: MSD Plots for Scenario 1 Data
- Supplementary Figure 2: MSD Plots for Scenario 2 Data
- Supplementary Figure 3: MSD Plots for Scenario 3 Data
- Supplementary Figure 4: MSD Plots for Scenario 4 Data
- Supplementary Figure 5: Velocity Plots for Scenario 1 Data
- Supplementary Figure 6: Velocity Plots for Scenario 2 Data
- Supplementary Figure 7: Velocity Plots for Scenario 3 Data
- Supplementary Figure 8: Velocity Plots for Scenario 4 Data
- Supplementary Figure 9: Localization Plots for Scenario 1 Data
- Supplementary Figure 10: Localization Plots for Scenario 2 Data
- Supplementary Figure 11: Localization Plots for Scenario 3 Data
- Supplementary Figure 12: Localization Plots for Scenario 4 Data
- Supplementary Table 5: Top-3 Best Methods for Different Gates
- Supplementary Note 1: Particle Tracking Methods
- Supplementary Note 2: Simulated Image Data Sets
- Supplementary Note 3: Performance Measures
- Supplementary Note 4: Simulation and Evaluation Software Tools

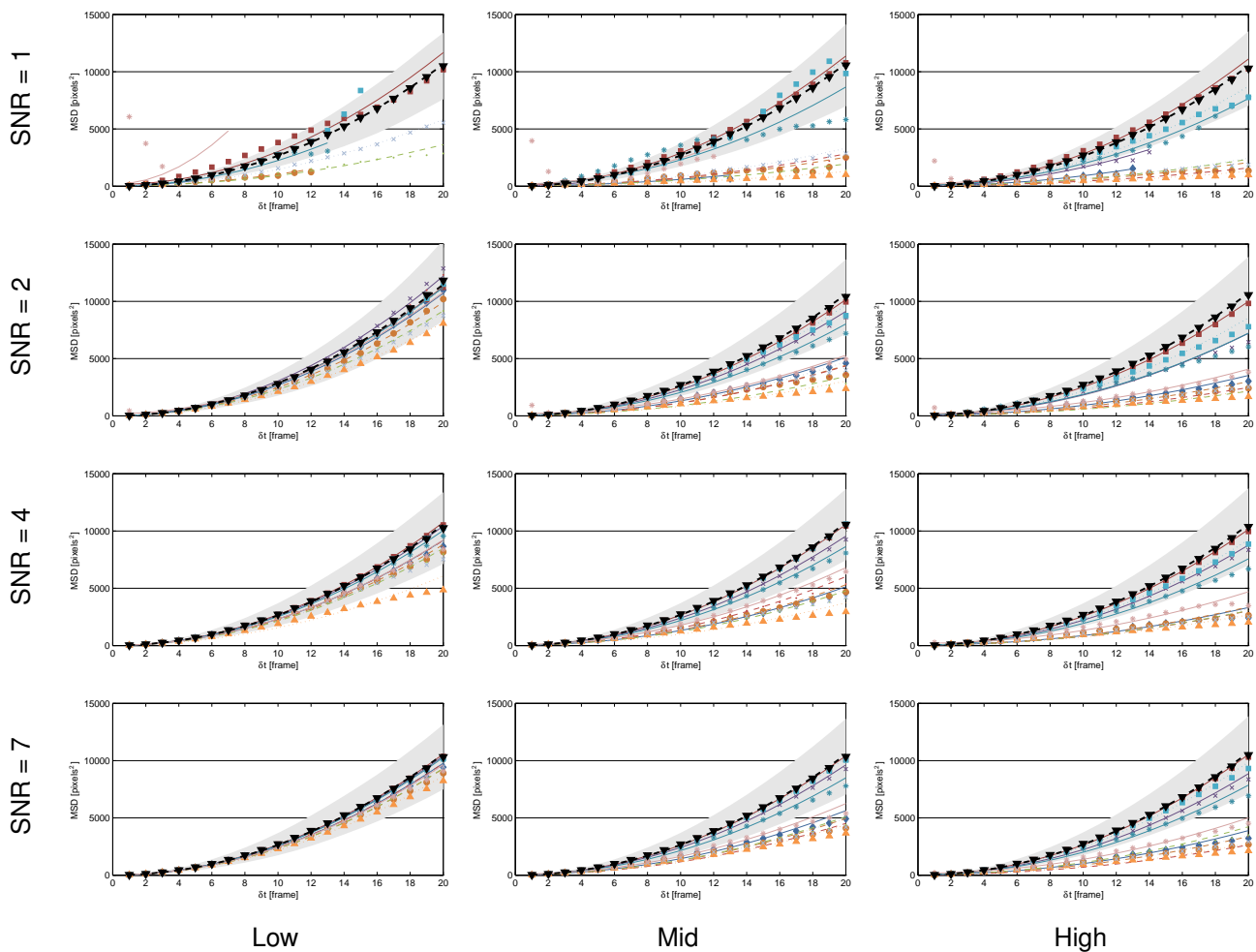
Supplementary Figure 1: MSD Plots for Scenario 1 Data



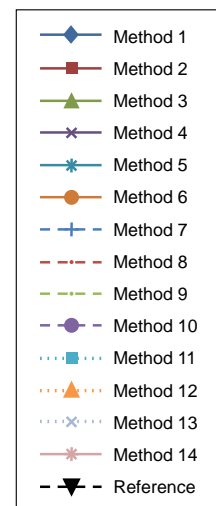
Mean-squared displacement (MSD) analysis results for the 12 image sequences of Scenario 1 in the competition data set. The graphs show the results for the four different SNR levels (1, 2, 4, 7) and three different density levels (Low, Mid, High). Each plot shows the results for all applicable particle tracking methods as well as the ground-truth (reference) data using the same color coding throughout (see legend). The MSD curves are shown for time intervals ranging from 1 to 20 frames (the upper time-interval limit was approximately equal to the average track length in the data). The gray-shaded area in each plot indicates the uncertainty in the ground-truth MSD values (from minus to plus one time the standard deviation around the plotted mean MSD value). For each method, the corresponding MSD values are indicated by points (using a different point shape and color for each method), and the result of curve fitting is indicated by a line drawn through these points (using a different line style per method but the same color as the corresponding points). For the curve fitting in this scenario (Brownian motion) we used linear functions. A weighted fitting was used (based on the standard deviation of MSD values) to account for the difference in reliability for different time lags (reflected by the gray-shaded area). The fitting was done using Matlab (Version 8.1.0.604 / R2013a, MathWorks, Natick, MA, USA). Note that in some cases it was not possible to find a good fit, resulting in anomalous curves. And for some methods, no MSD values could be computed for the longer time intervals, because apparently the method produced only fragmented tracks of relatively short length.



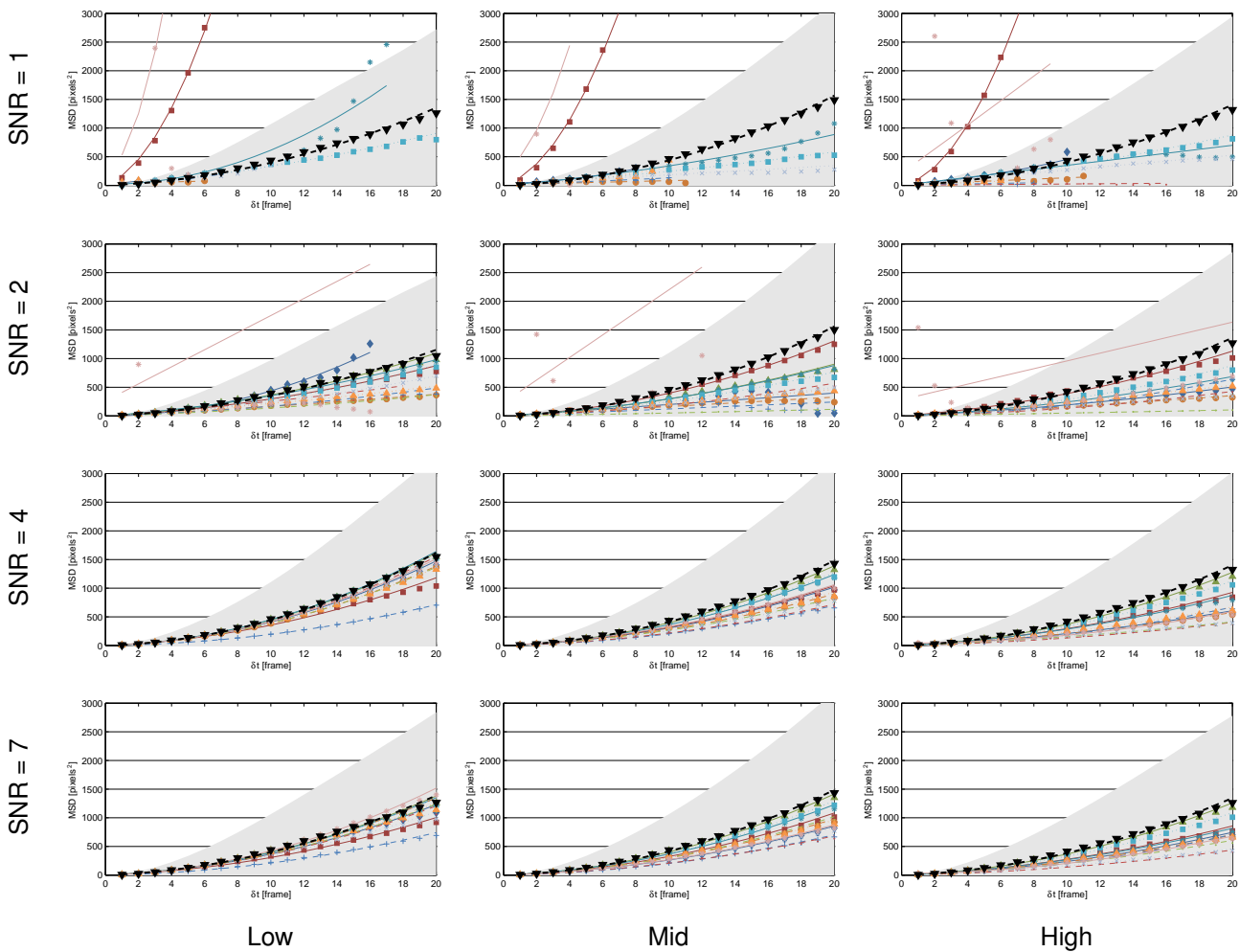
Supplementary Figure 2: MSD Plots for Scenario 2 Data



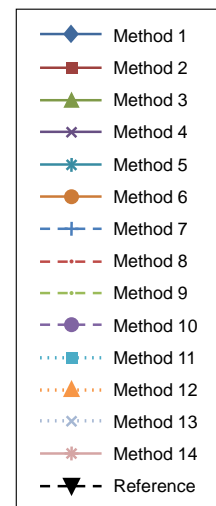
Mean-squared displacement (MSD) analysis results for the 12 image sequences of Scenario 2 in the competition data set. The graphs show the results for the four different SNR levels (1, 2, 4, 7) and three different density levels (Low, Mid, High). Each plot shows the results for all applicable particle tracking methods as well as the ground-truth (reference) data using the same color coding throughout (see legend). The MSD curves are shown for time intervals ranging from 1 to 20 frames (the upper time-interval limit was approximately equal to the average track length in the data). The gray-shaded area in each plot indicates the uncertainty in the ground-truth MSD values (from minus to plus one time the standard deviation around the plotted mean MSD value). For each method, the corresponding MSD values are indicated by points (using a different point shape and color for each method), and the result of curve fitting is indicated by a line drawn through these points (using a different line style per method but the same color as the corresponding points). For the curve fitting in this scenario (directed motion) we used quadratic functions. A weighted fitting was used (based on the standard deviation of MSD values) to account for the difference in reliability for different time lags (reflected by the gray-shaded area). The fitting was done using Matlab (Version 8.1.0.604 / R2013a, MathWorks, Natick, MA, USA). Note that in some cases it was not possible to find a good fit, resulting in anomalous curves. And for some methods, no MSD values could be computed for the longer time intervals, because apparently the method produced only fragmented tracks of relatively short length.



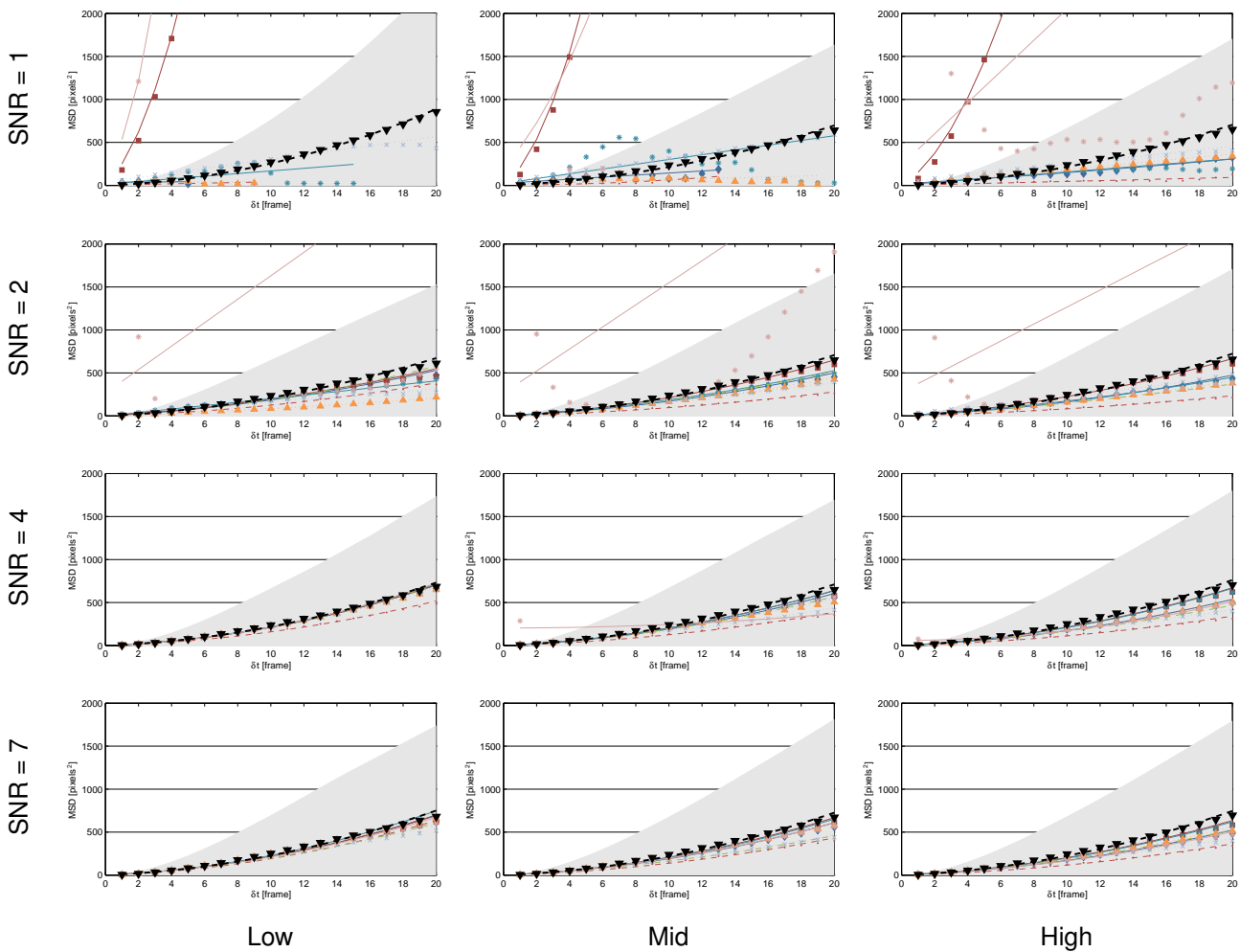
Supplementary Figure 3: MSD Plots for Scenario 3 Data



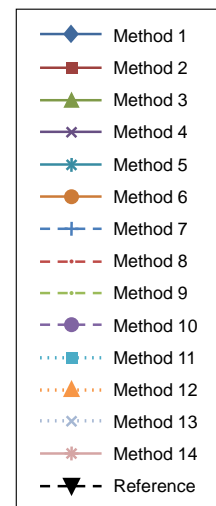
Mean-squared displacement (MSD) analysis results for the 12 image sequences of Scenario 3 in the competition data set. The graphs show the results for the four different SNR levels (1, 2, 4, 7) and three different density levels (Low, Mid, High). Each plot shows the results for all applicable particle tracking methods as well as the ground-truth (reference) data using the same color coding throughout (see legend). The MSD curves are shown for time intervals ranging from 1 to 20 frames (the upper time-interval limit was approximately equal to the average track length in the data). The gray-shaded area in each plot indicates the uncertainty in the ground-truth MSD values (from minus to plus one time the standard deviation around the plotted mean MSD value). For each method, the corresponding MSD values are indicated by points (using a different point shape and color for each method), and the result of curve fitting is indicated by a line drawn through these points (using a different line style per method but the same color as the corresponding points). For the curve fitting in this scenario (combined Brownian and directed motion) we used linear + quadratic functions. A weighted fitting was used (based on the standard deviation of MSD values) to account for the difference in reliability for different time lags (reflected by the gray-shaded area). The fitting was done using Matlab (Version 8.1.0.604 / R2013a, MathWorks, Natick, MA, USA). Note that in some cases it was not possible to find a good fit, resulting in anomalous curves. And for some methods, no MSD values could be computed for the longer time intervals, because apparently the method produced only fragmented tracks of relatively short length.



Supplementary Figure 4: MSD Plots for Scenario 4 Data



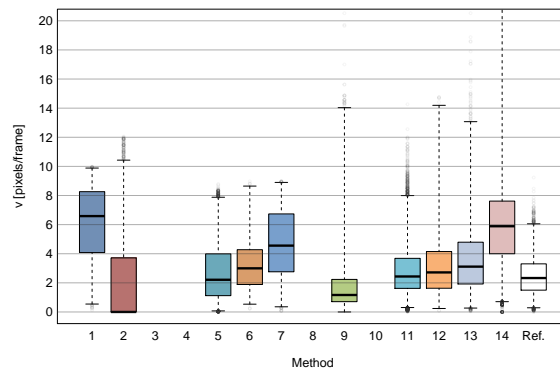
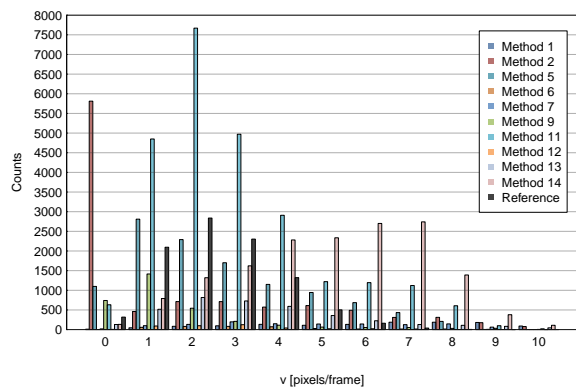
Mean-squared displacement (MSD) analysis results for the 12 image sequences of Scenario 4 in the competition data set. The graphs show the results for the four different SNR levels (1, 2, 4, 7) and three different density levels (Low, Mid, High). Each plot shows the results for all applicable particle tracking methods as well as the ground-truth (reference) data using the same color coding throughout (see legend). The MSD curves are shown for time intervals ranging from 1 to 20 frames (the upper time-interval limit was approximately equal to the average track length in the data). The gray-shaded area in each plot indicates the uncertainty in the ground-truth MSD values (from minus to plus one time the standard deviation around the plotted mean MSD value). For each method, the corresponding MSD values are indicated by points (using a different point shape and color for each method), and the result of curve fitting is indicated by a line drawn through these points (using a different line style per method but the same color as the corresponding points). For the curve fitting in this scenario (combined Brownian and directed motion) we used linear + quadratic functions. A weighted fitting was used (based on the standard deviation of MSD values) to account for the difference in reliability for different time lags (reflected by the gray-shaded area). The fitting was done using Matlab (Version 8.1.0.604 / R2013a, MathWorks, Natick, MA, USA). Note that in some cases it was not possible to find a good fit, resulting in anomalous curves. And for some methods, no MSD values could be computed for the longer time intervals, because apparently the method produced only fragmented tracks of relatively short length.



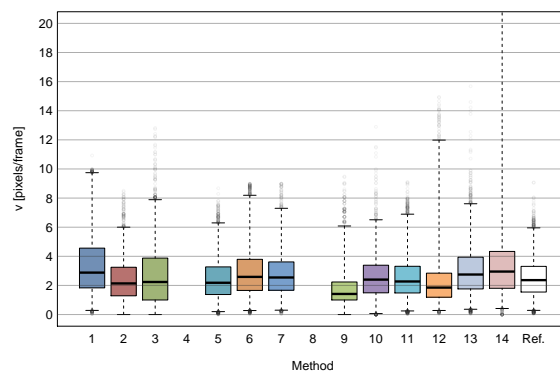
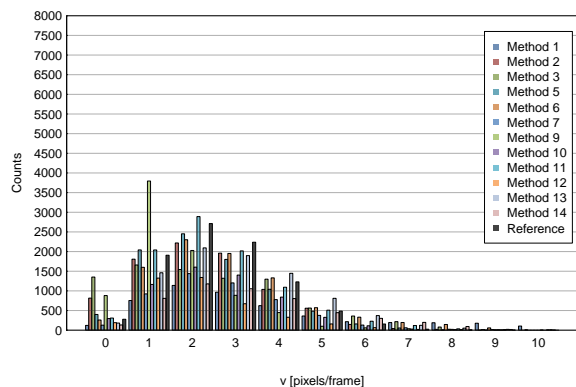
Supplementary Figure 5: Velocity Plots for Scenario 1 Data

The following pages show the velocity histograms (left plots) and corresponding box plots (right plots) for the 12 image sequences of Scenario 1 in the competition data set. Each page shows the results for one density level (Low, Mid, High) and the four different SNR levels (1, 2, 4, 7). Each plot shows the results for all applicable particle tracking methods as well as the ground-truth (reference) data using the same color coding throughout. In the box plots, the boxes indicate the 25th-75th percentile, the horizontal bar inside a box indicates the median value, the whiskers range from the 1st to the 99th percentile, and the gray circles above/below the whiskers indicate the outliers.

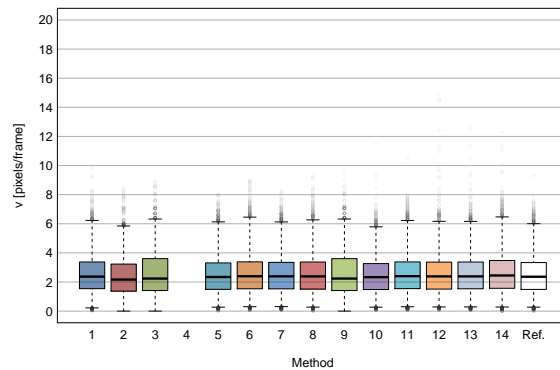
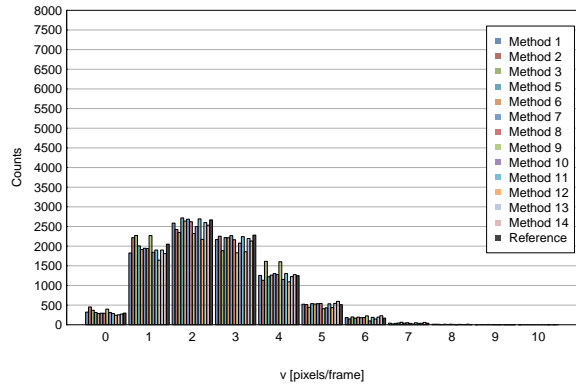
SNR = 1



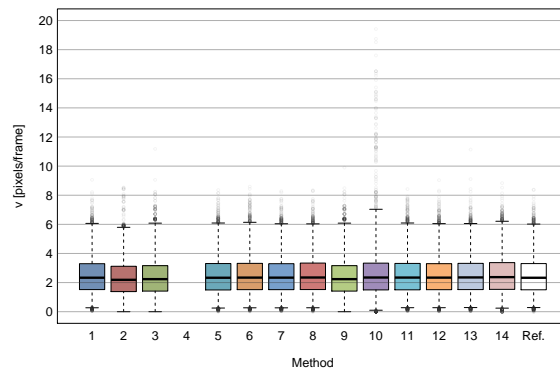
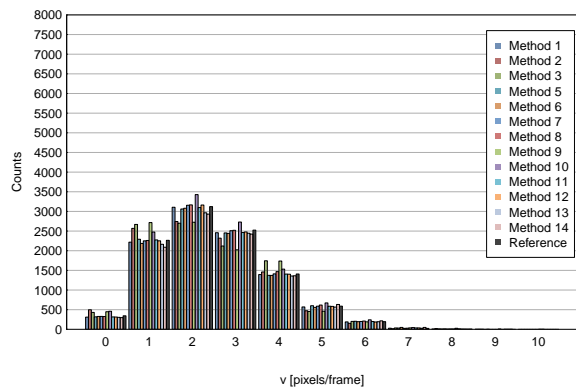
SNR = 2



SNR = 4

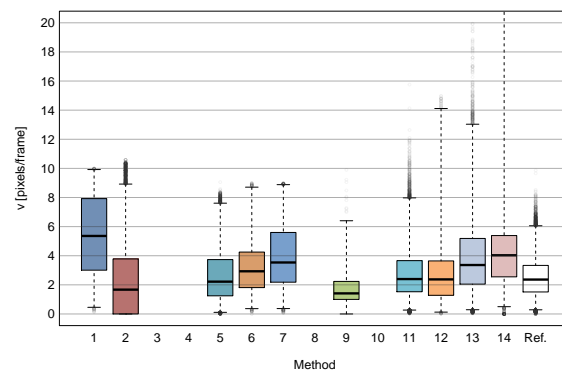
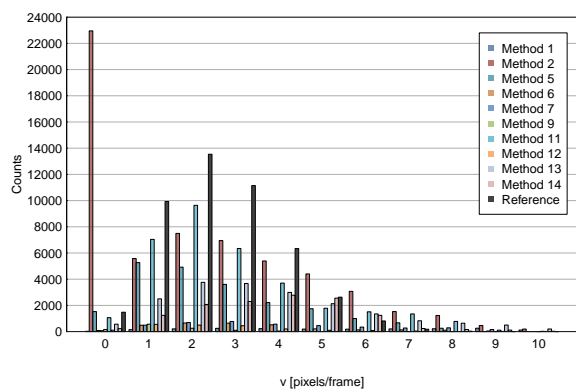


SNR = 7

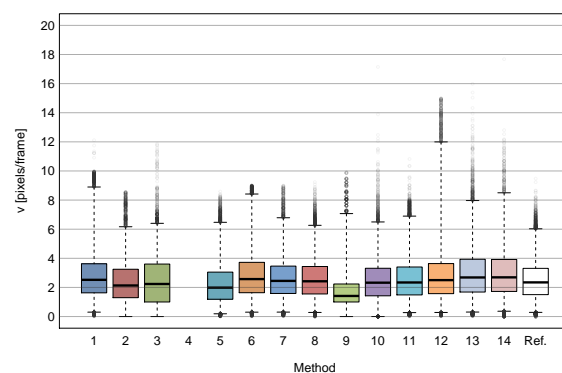
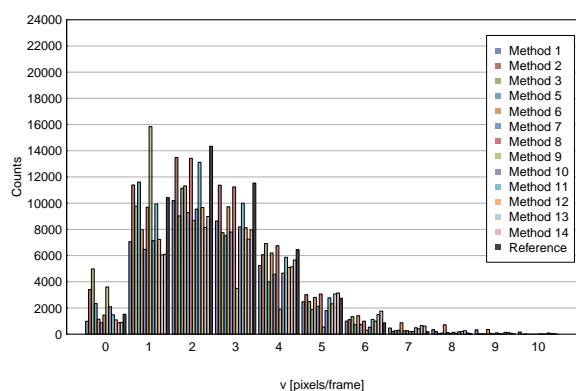


Scenario 1 Low Density

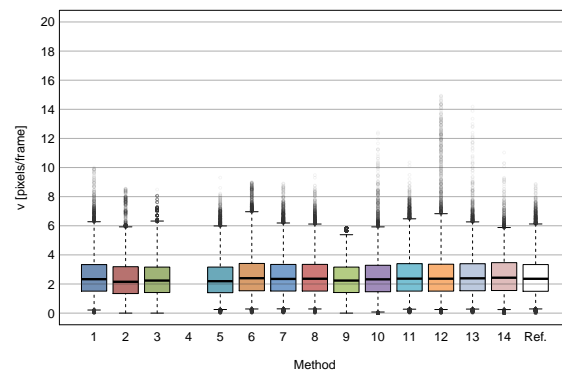
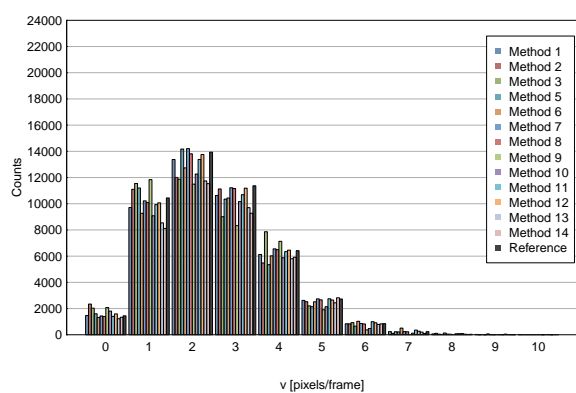
SNR = 1



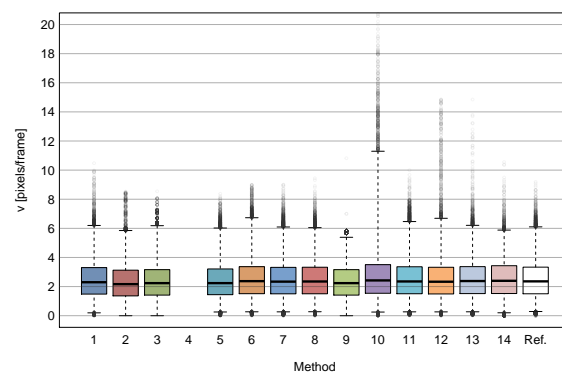
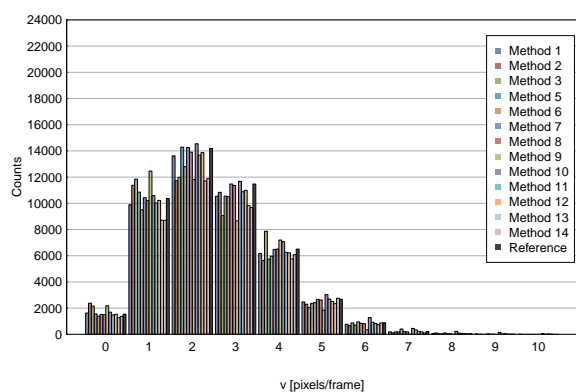
SNR = 2



SNR = 4

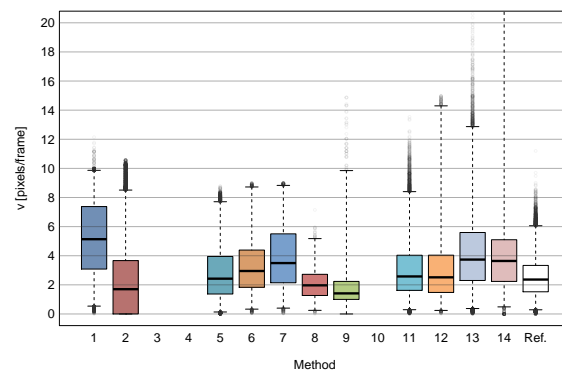
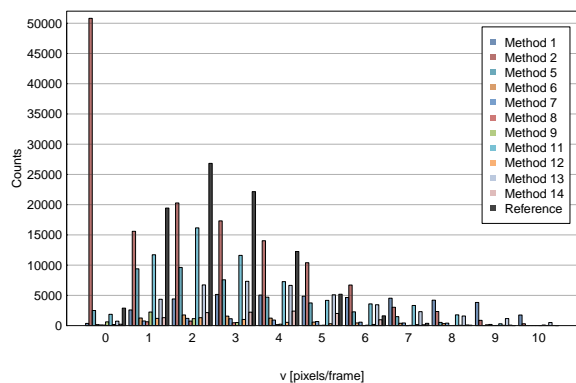


SNR = 7

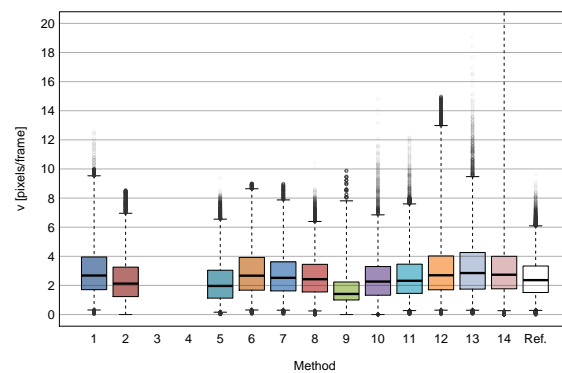
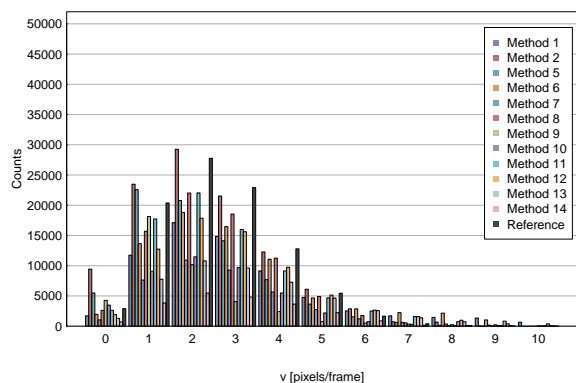


Scenario 1 Mid Density

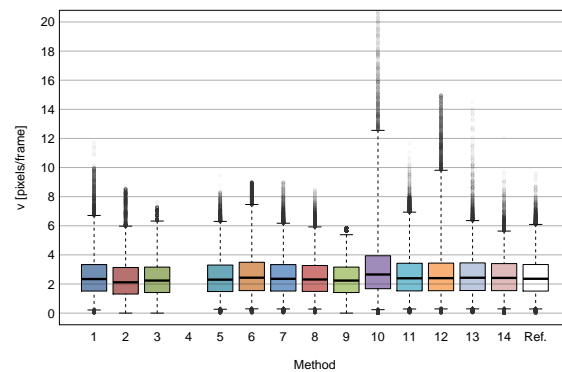
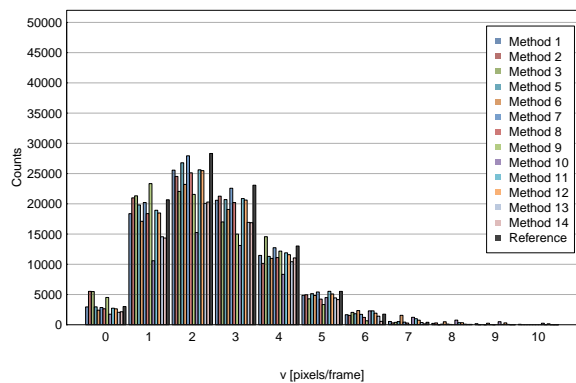
SNR = 1



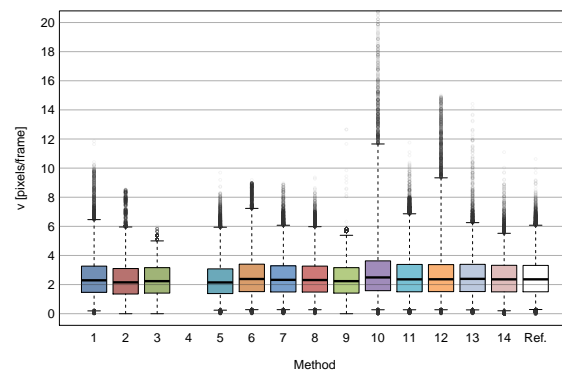
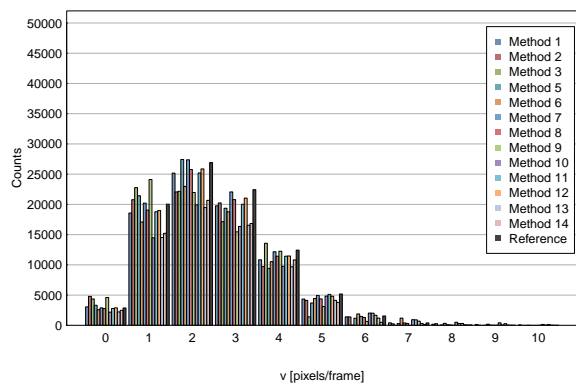
SNR = 2



SNR = 4



SNR = 7

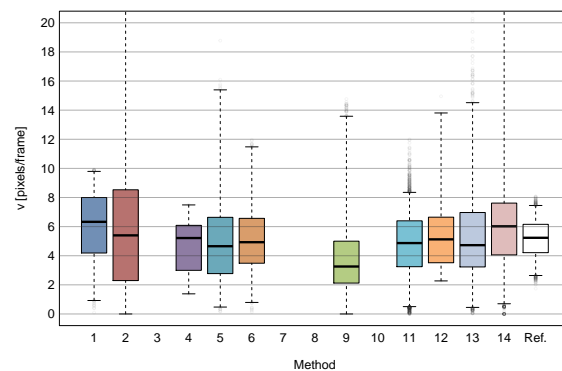
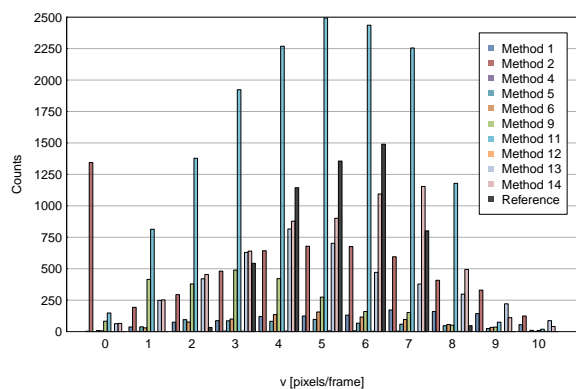


Scenario 1 High Density

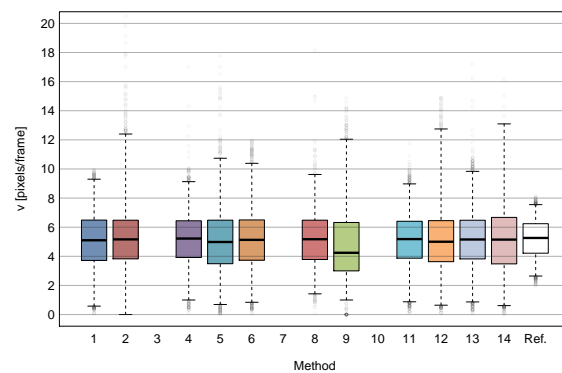
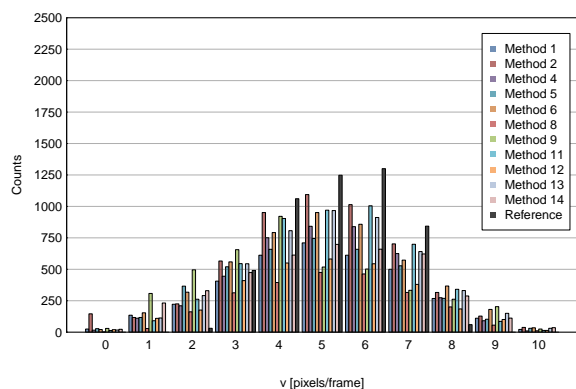
Supplementary Figure 6: Velocity Plots for Scenario 2 Data

The following pages show the velocity histograms (left plots) and corresponding box plots (right plots) for the 12 image sequences of Scenario 2 in the competition data set. Each page shows the results for one density level (Low, Mid, High) and the four different SNR levels (1, 2, 4, 7). Each plot shows the results for all applicable particle tracking methods as well as the ground-truth (reference) data using the same color coding throughout. In the box plots, the boxes indicate the 25th-75th percentile, the horizontal bar inside a box indicates the median value, the whiskers range from the 1st to the 99th percentile, and the gray circles above/below the whiskers indicate the outliers.

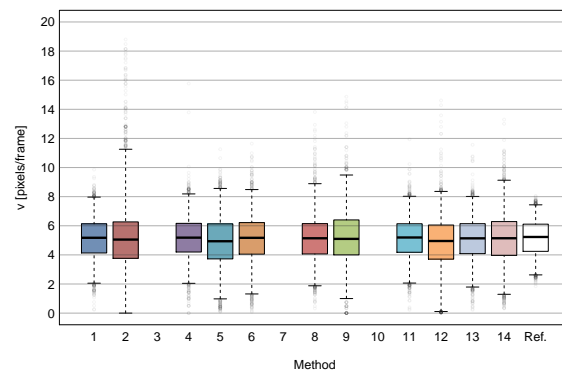
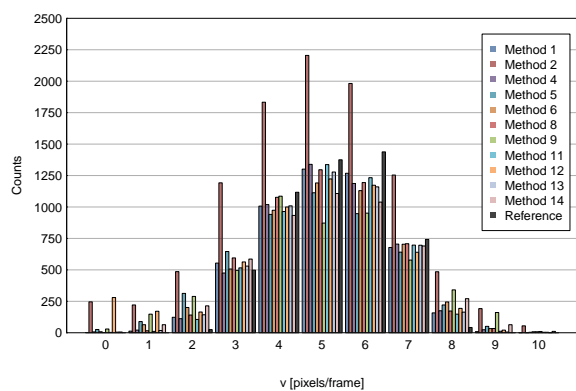
SNR = 1



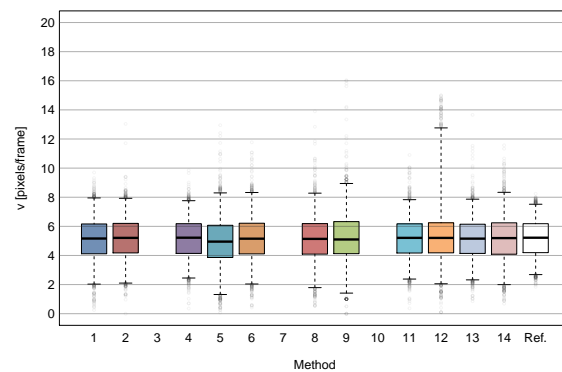
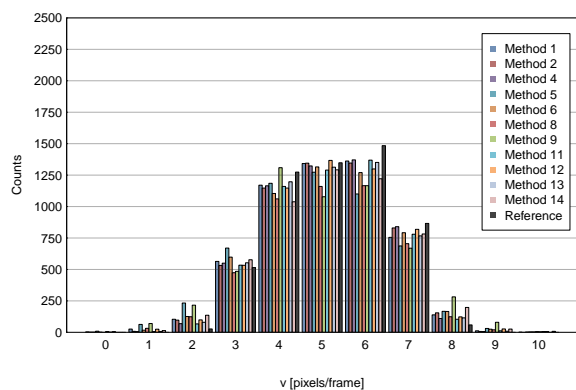
SNR = 2



SNR = 4

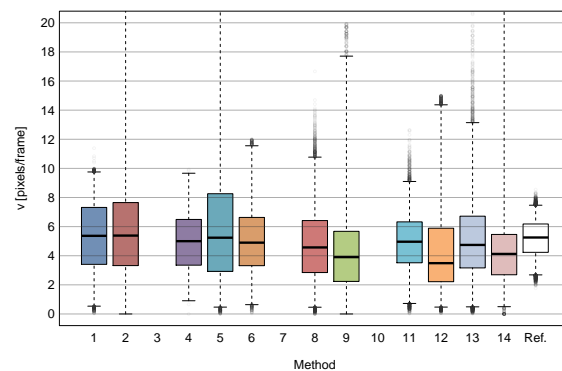
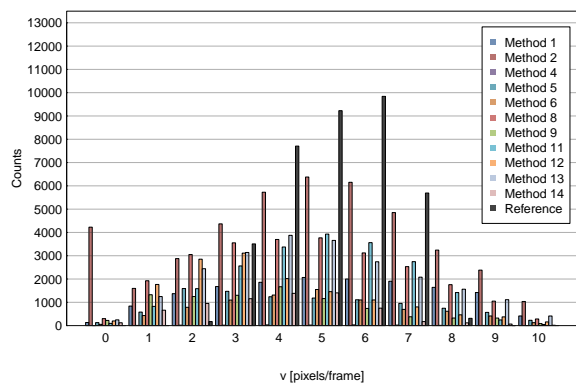


SNR = 7

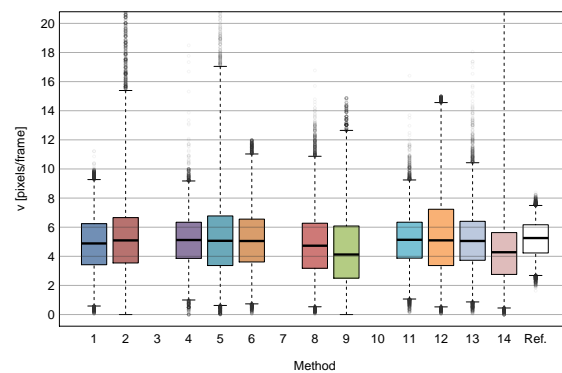
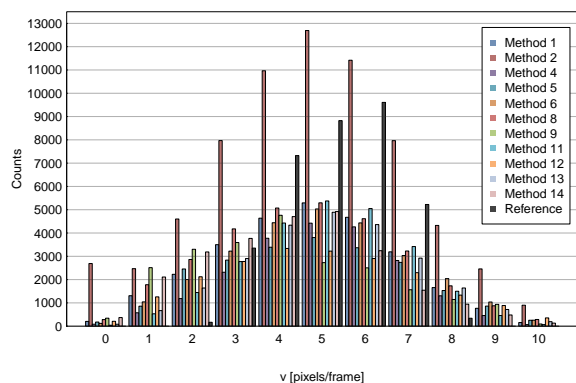


Scenario 2 Low Density

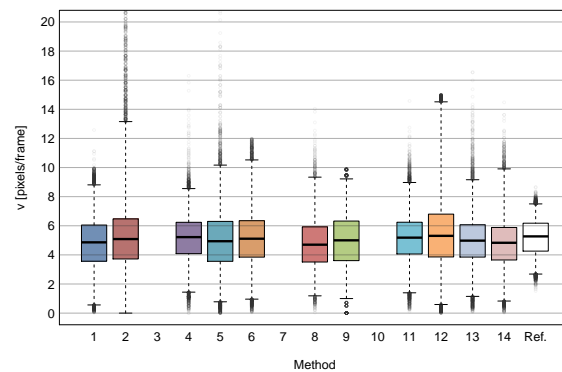
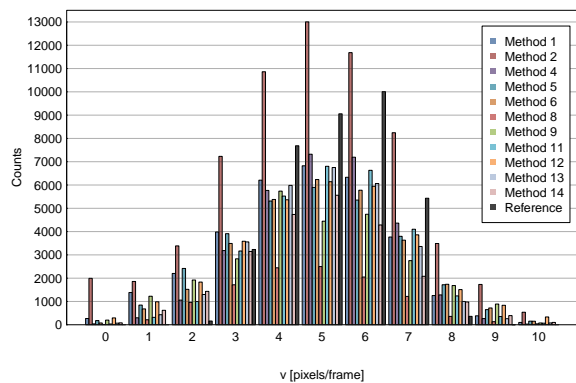
SNR = 1



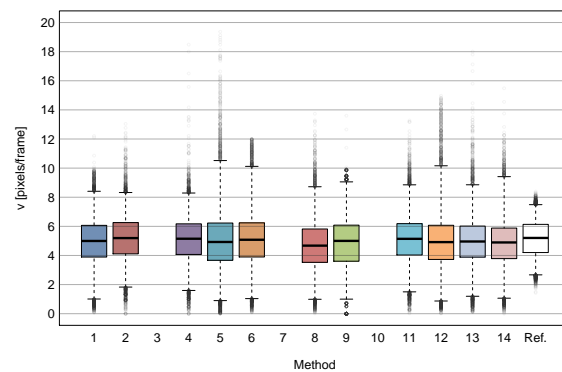
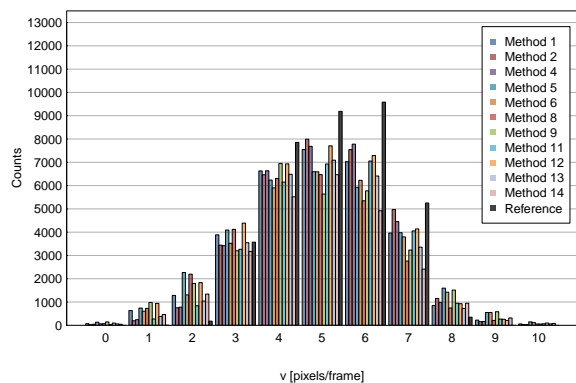
SNR = 2



SNR = 4

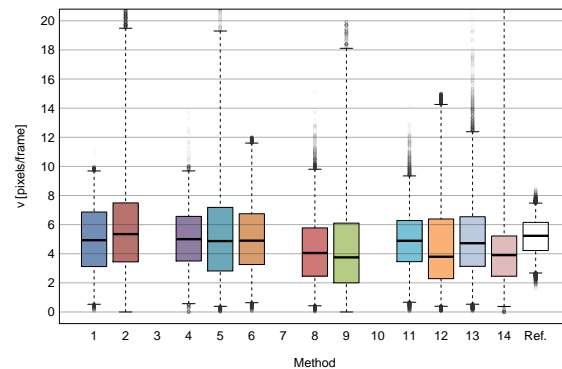
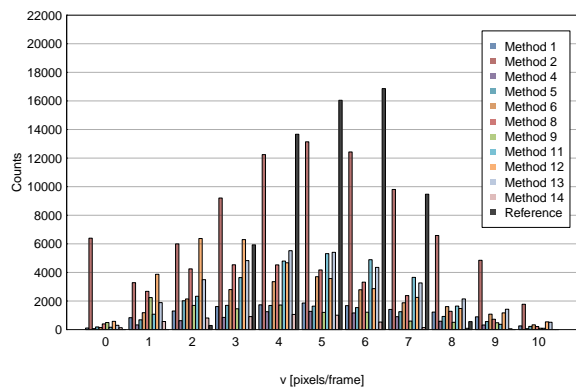


SNR = 7

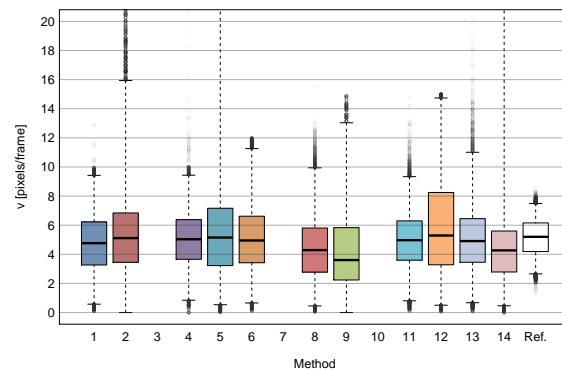
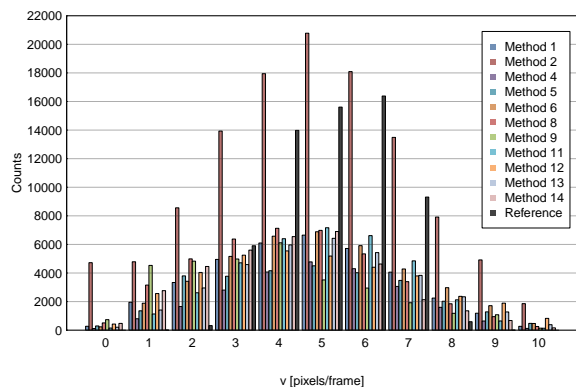


Scenario 2 Mid Density

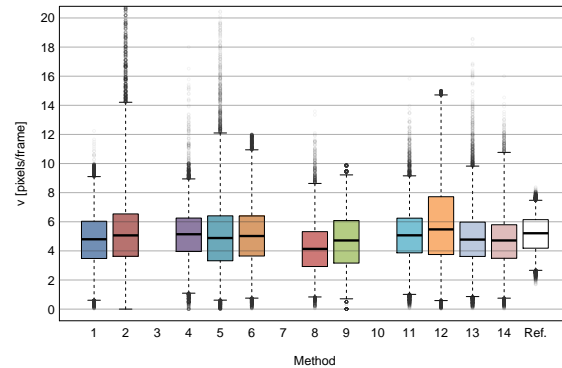
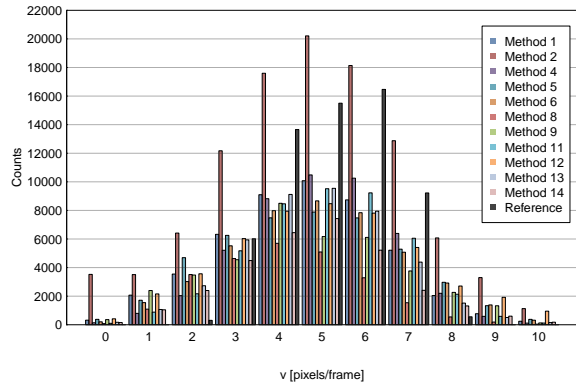
SNR = 1



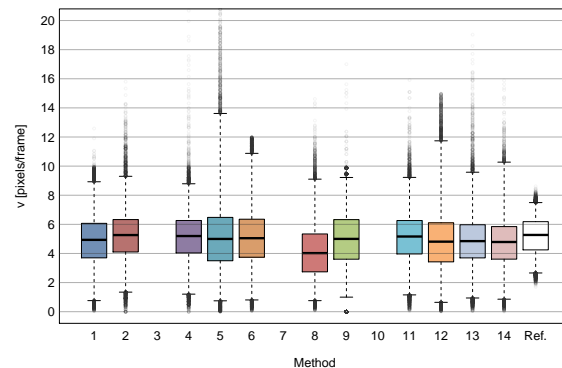
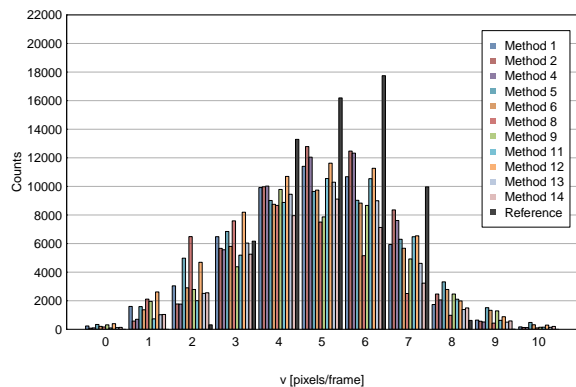
SNR = 2



SNR = 4



SNR = 7

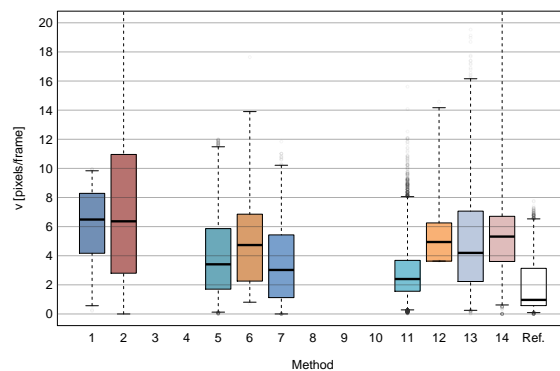
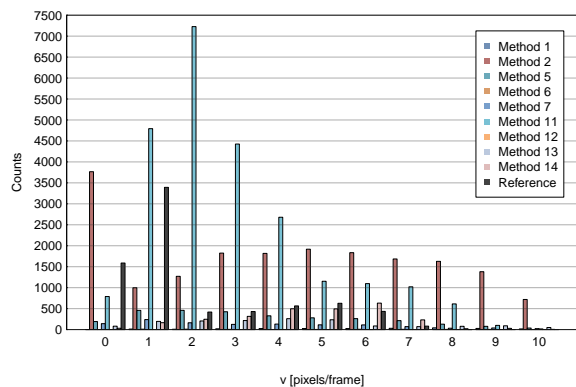


Scenario 2 High Density

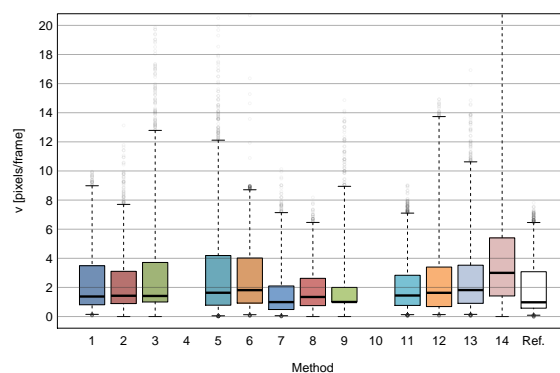
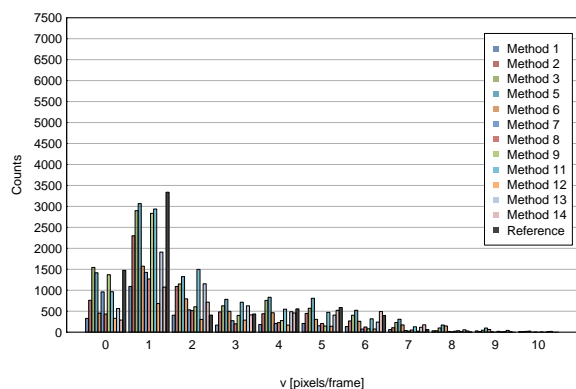
Supplementary Figure 7: Velocity Plots for Scenario 3 Data

The following pages show the velocity histograms (left plots) and corresponding box plots (right plots) for the 12 image sequences of Scenario 3 in the competition data set. Each page shows the results for one density level (Low, Mid, High) and the four different SNR levels (1, 2, 4, 7). Each plot shows the results for all applicable particle tracking methods as well as the ground-truth (reference) data using the same color coding throughout. In the box plots, the boxes indicate the 25th-75th percentile, the horizontal bar inside a box indicates the median value, the whiskers range from the 1st to the 99th percentile, and the gray circles above/below the whiskers indicate the outliers.

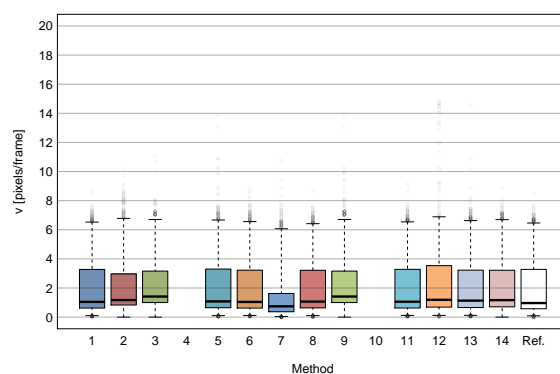
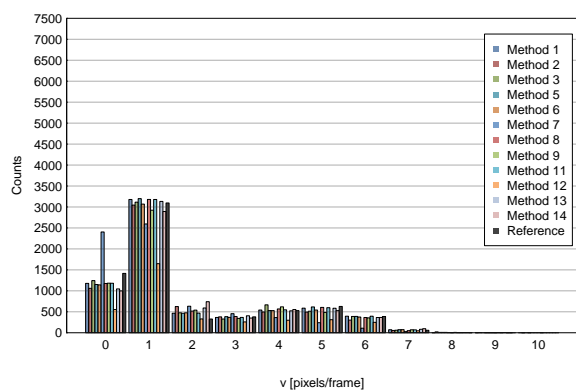
SNR = 1



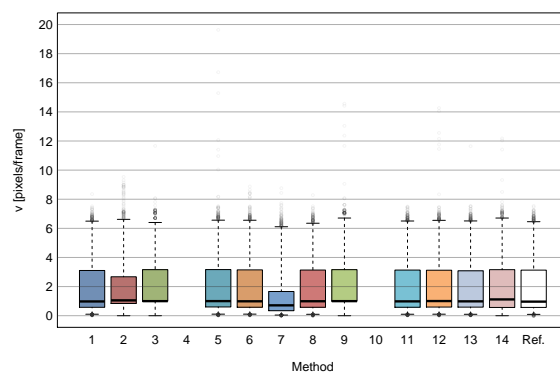
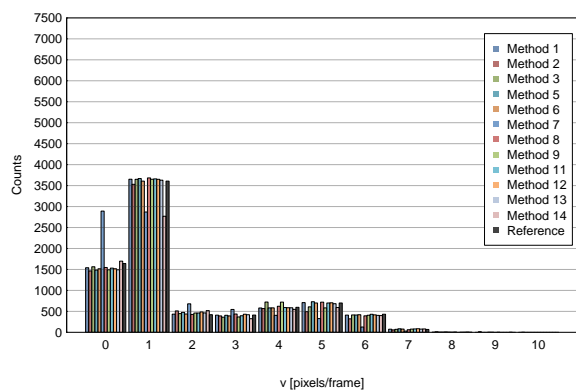
SNR = 2



SNR = 4

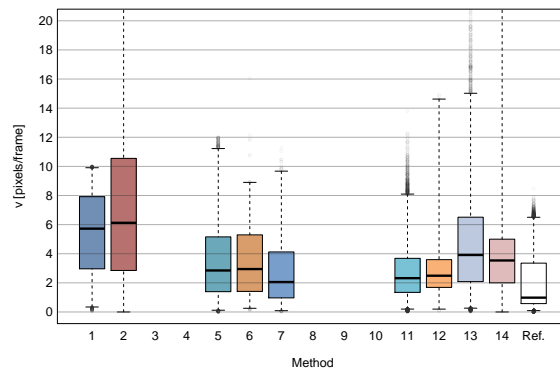
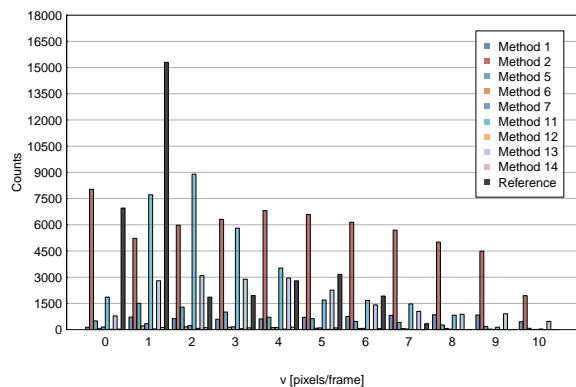


SNR = 7

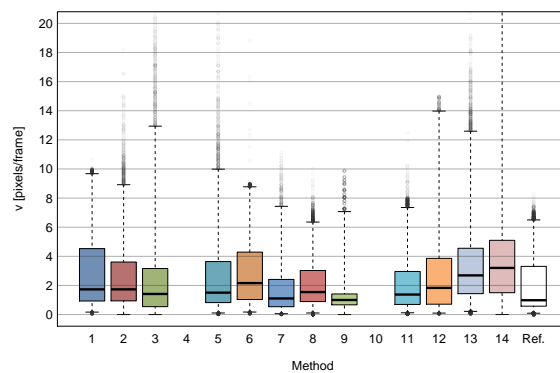
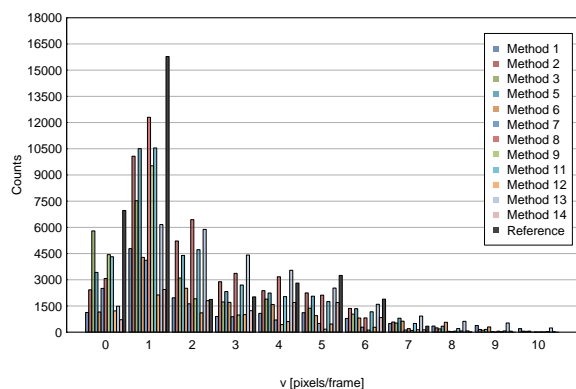


Scenario 3 Low Density

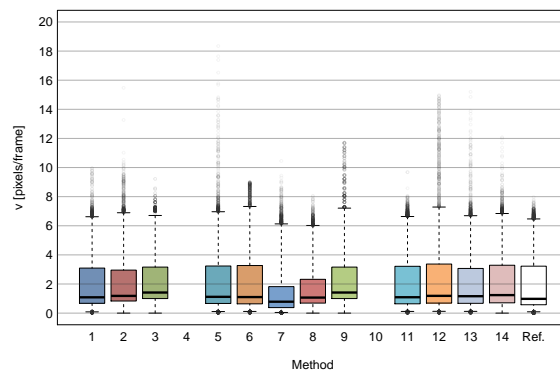
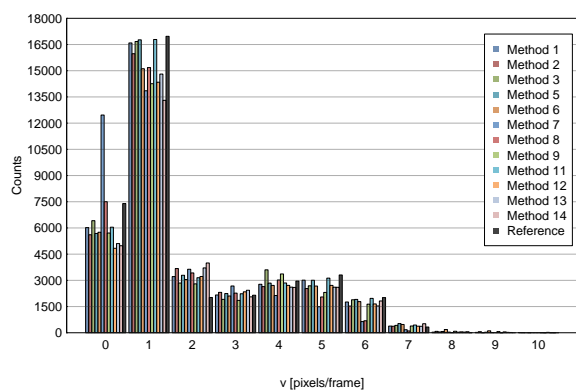
SNR = 1



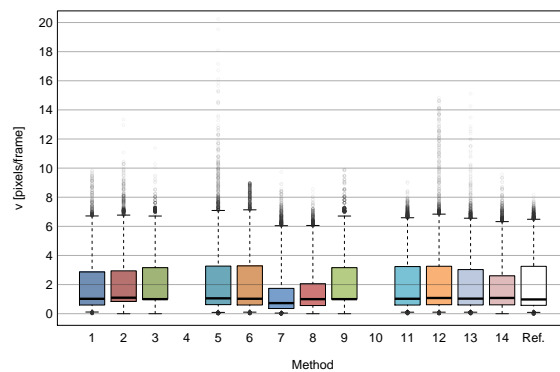
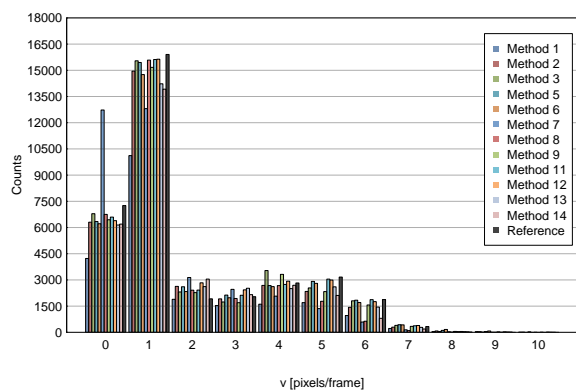
SNR = 2



SNR = 4

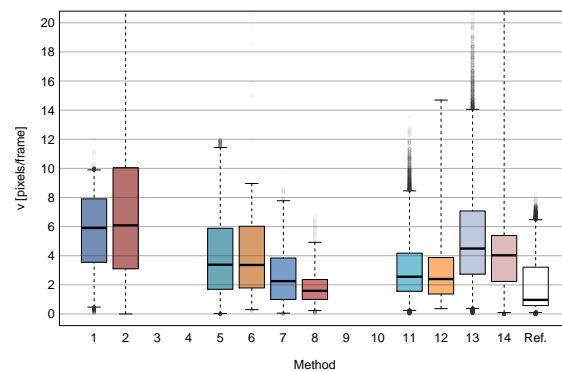
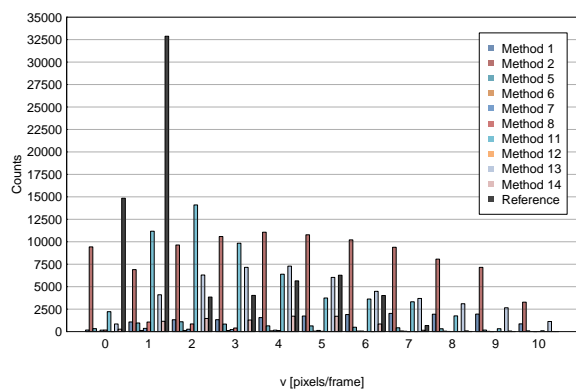


SNR = 7

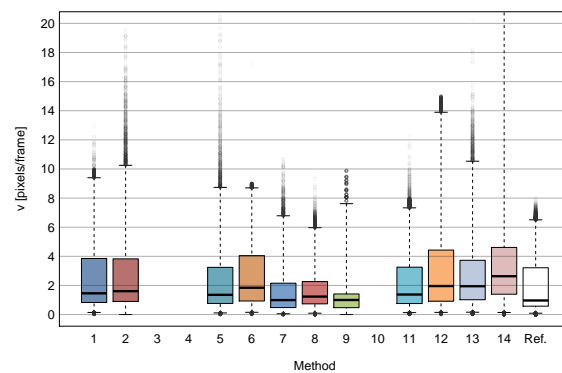
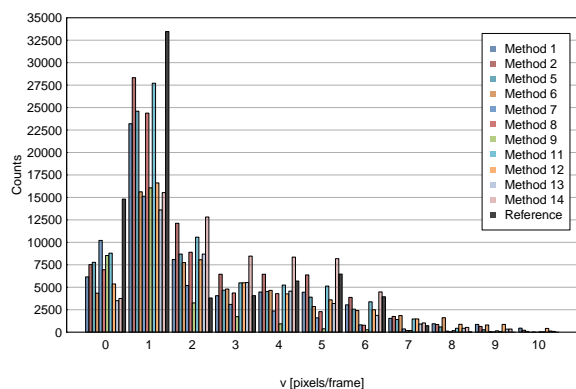


Scenario 3 Mid Density

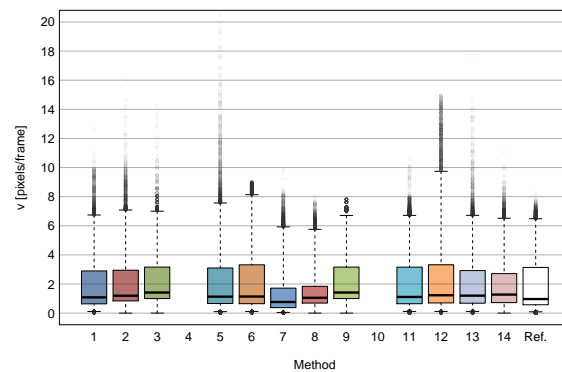
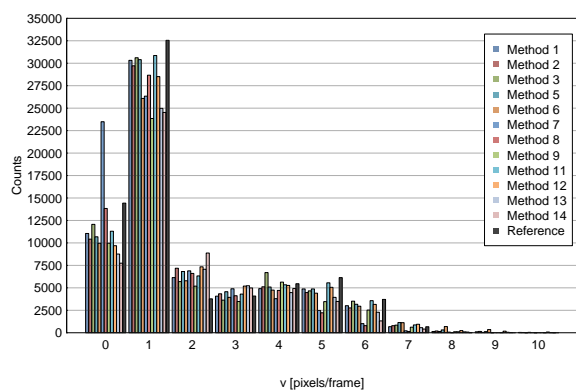
SNR = 1



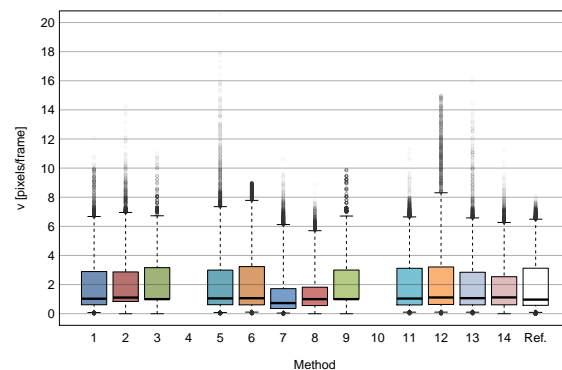
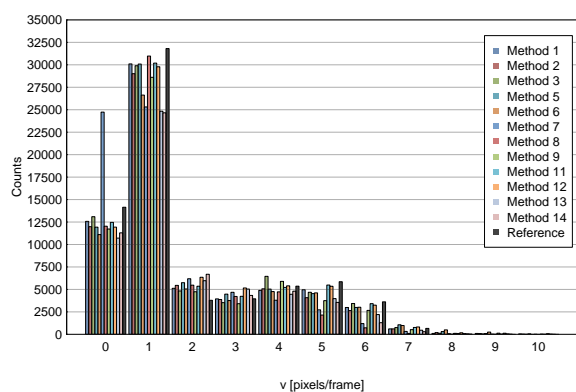
SNR = 2



SNR = 4



SNR = 7

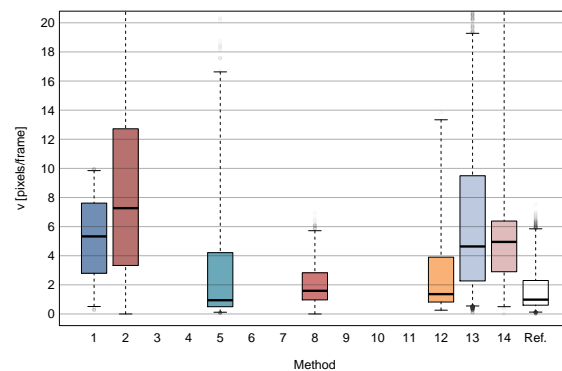
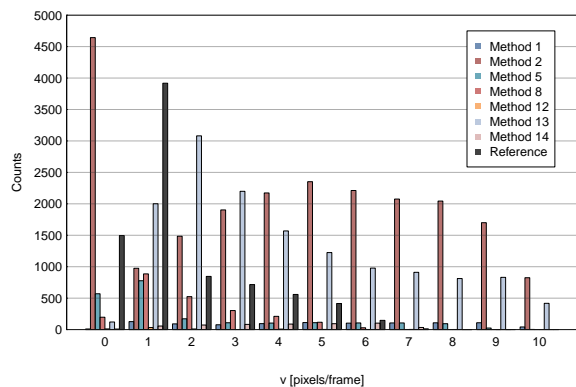


Scenario 3 High Density

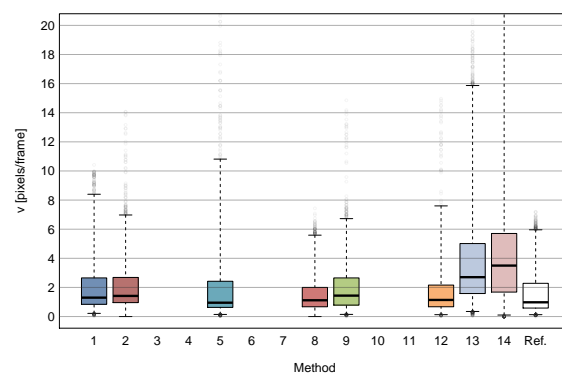
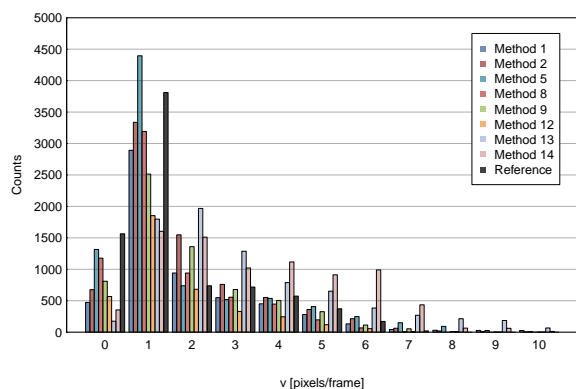
Supplementary Figure 8: Velocity Plots for Scenario 4 Data

The following pages show the velocity histograms (left plots) and corresponding box plots (right plots) for the 12 image sequences of Scenario 4 in the competition data set. Each page shows the results for one density level (Low, Mid, High) and the four different SNR levels (1, 2, 4, 7). Each plot shows the results for all applicable particle tracking methods as well as the ground-truth (reference) data using the same color coding throughout. In the box plots, the boxes indicate the 25th-75th percentile, the horizontal bar inside a box indicates the median value, the whiskers range from the 1st to the 99th percentile, and the gray circles above/below the whiskers indicate the outliers.

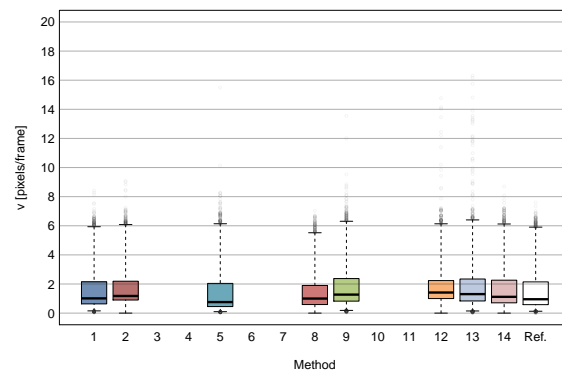
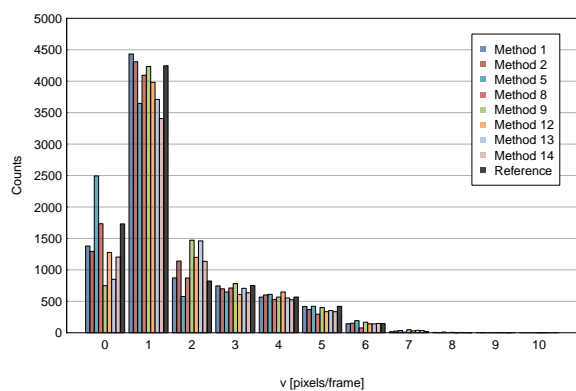
SNR = 1



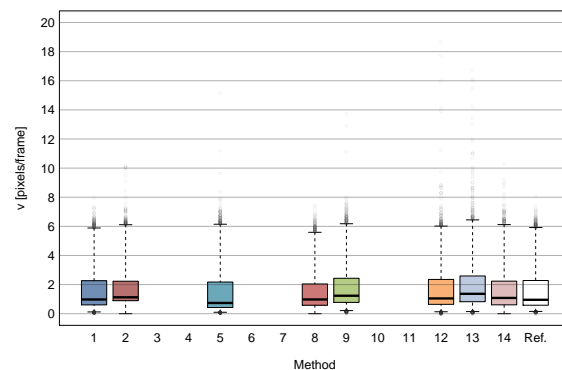
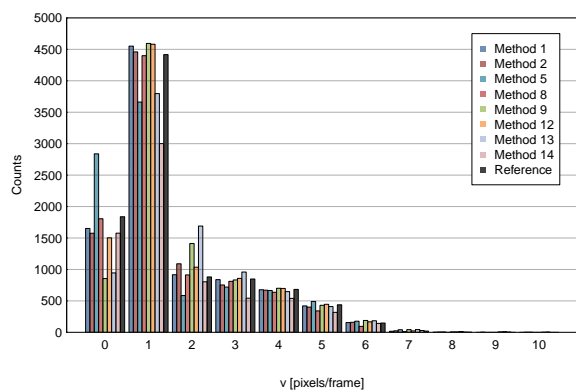
SNR = 2



SNR = 4

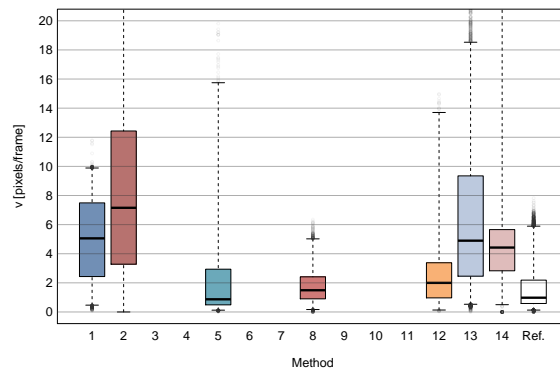
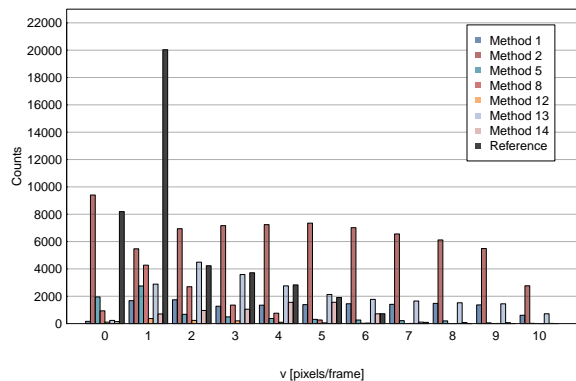


SNR = 7

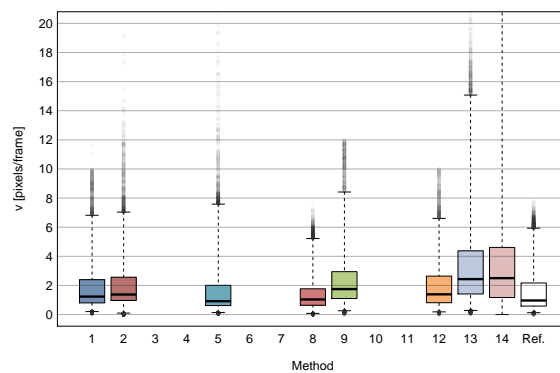
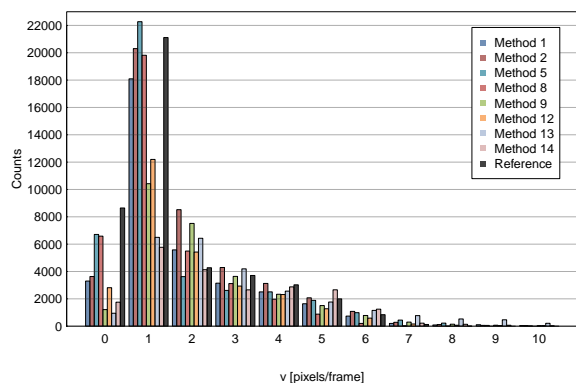


Scenario 4 Low Density

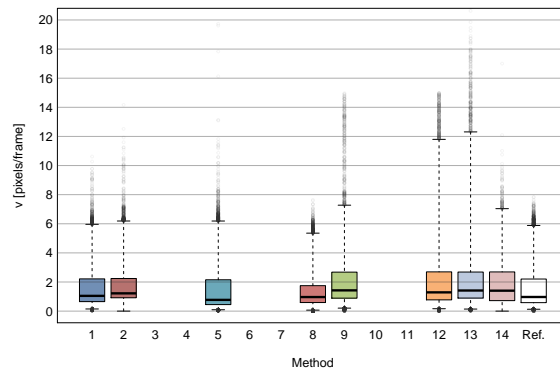
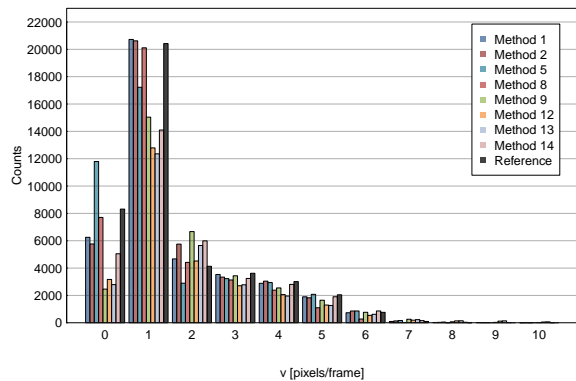
SNR = 1



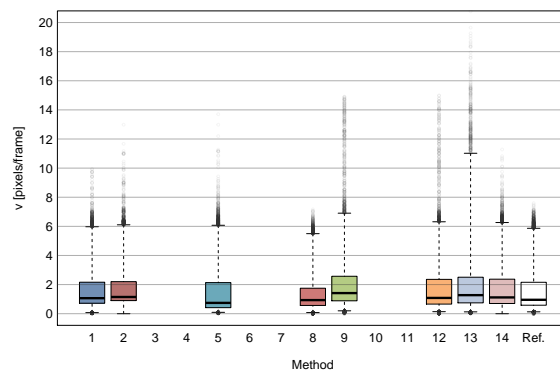
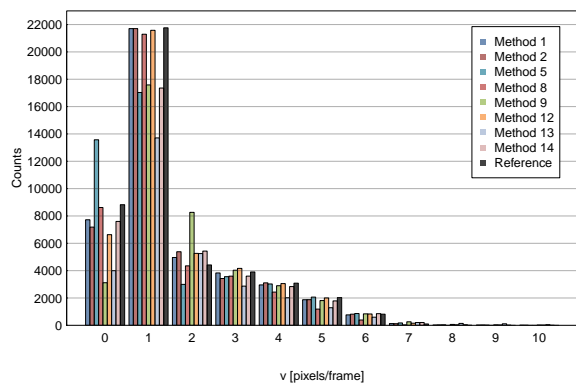
SNR = 2



SNR = 4

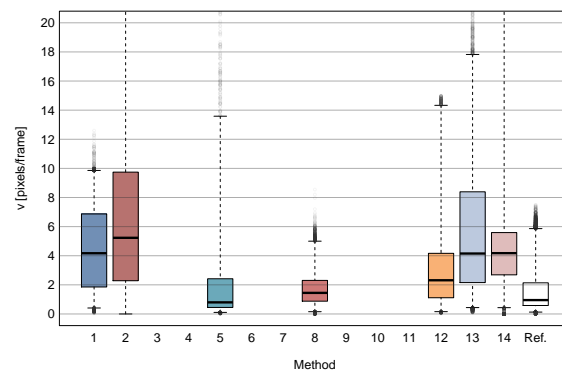
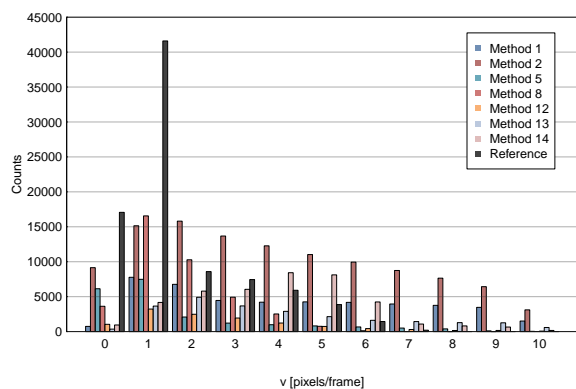


SNR = 7

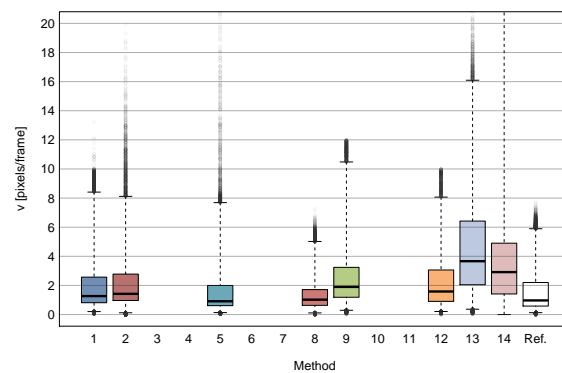
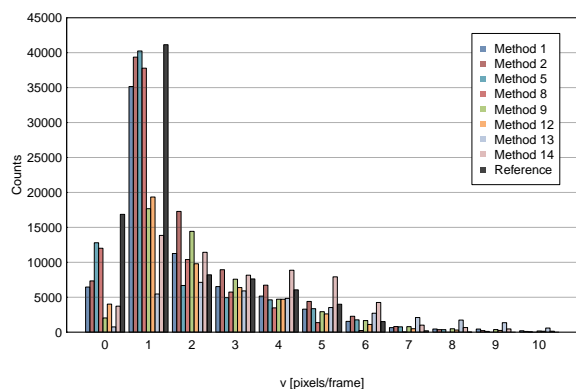


Scenario 4 Mid Density

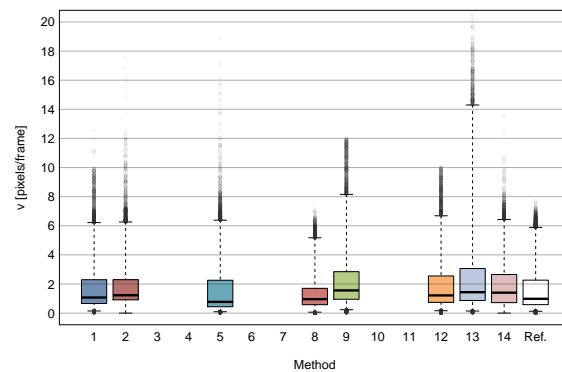
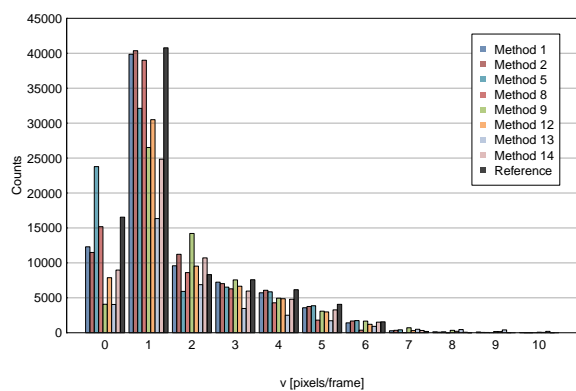
SNR = 1



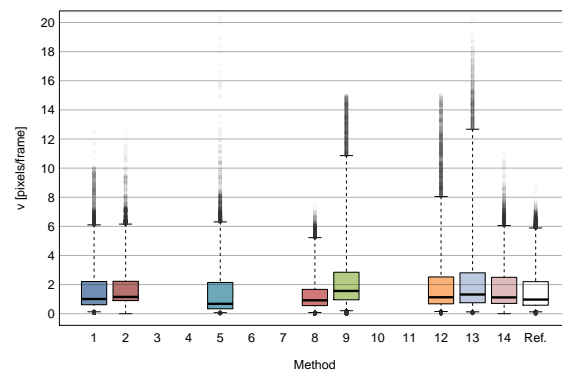
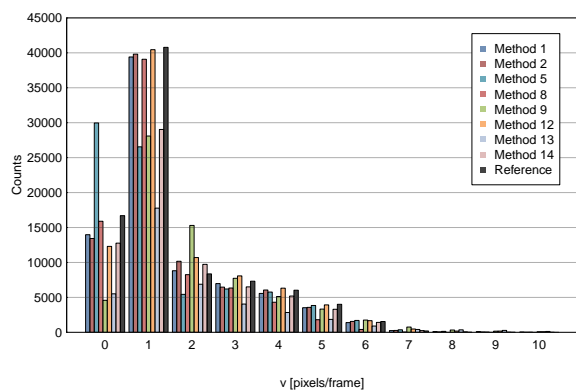
SNR = 2



SNR = 4



SNR = 7

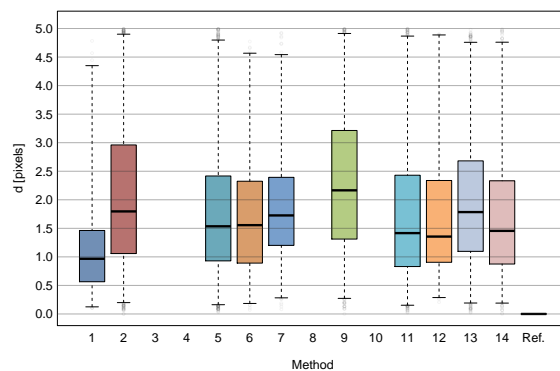
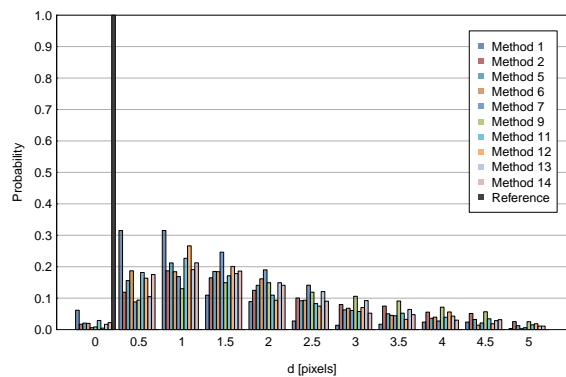


Scenario 4 High Density

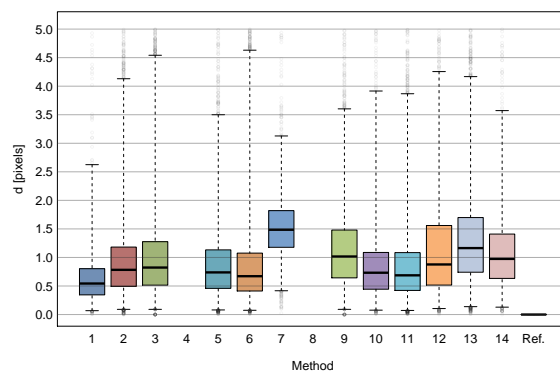
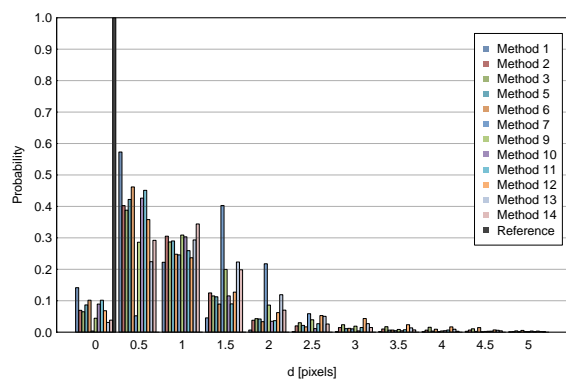
Supplementary Figure 9: Localization Plots for Scenario 1 Data

The following pages show the (normalized) localization histograms (left plots) and corresponding box plots (right plots) for the 12 image sequences of Scenario 1 in the competition data set. Each page shows the results for one density level (Low, Mid, High) and the four different SNR levels (1, 2, 4, 7). Each plot shows the results for all applicable particle tracking methods as well as the ground-truth (reference) data using the same color coding throughout. In the box plots, the boxes indicate the 25th-75th percentile, the horizontal bar inside a box indicates the median value, the whiskers range from the 1st to the 99th percentile, and the gray circles above/below the whiskers indicate the outliers.

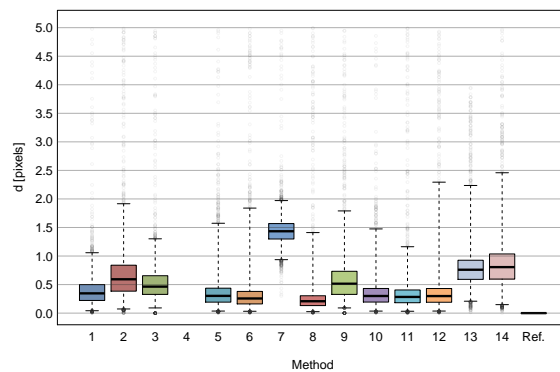
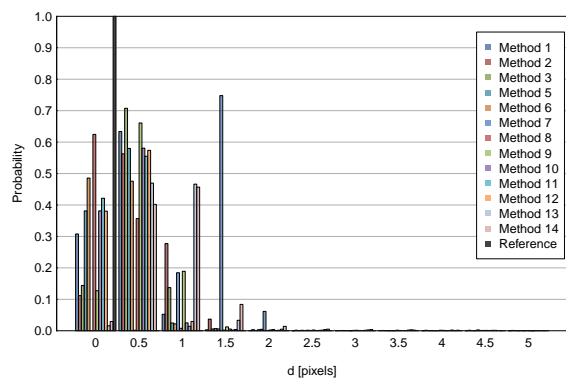
SNR = 1



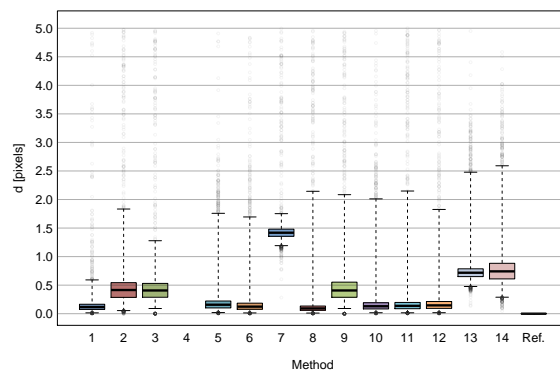
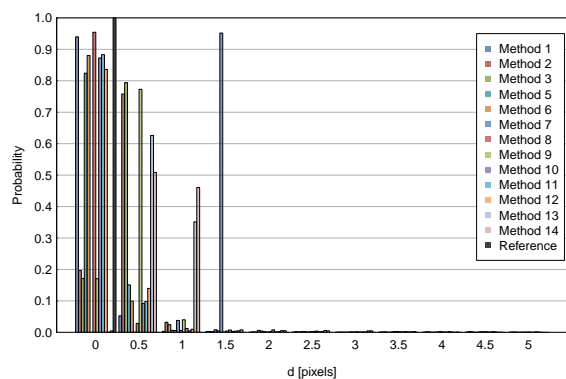
SNR = 2



SNR = 4

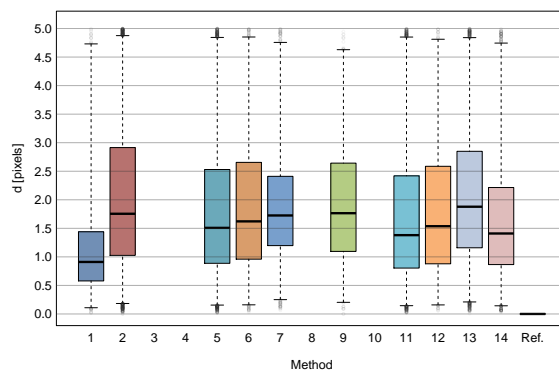
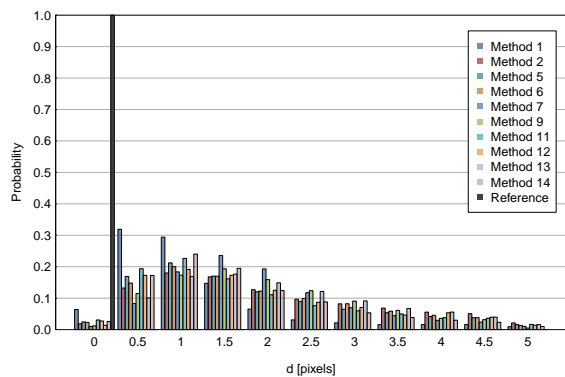


SNR = 7

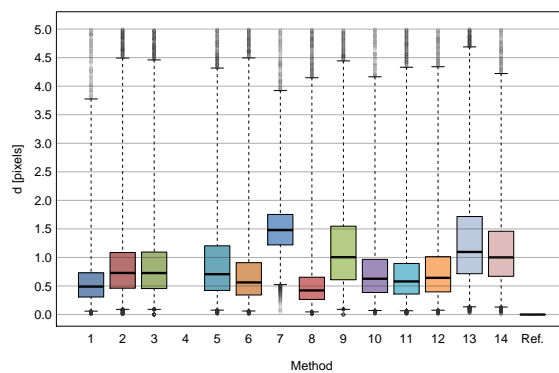
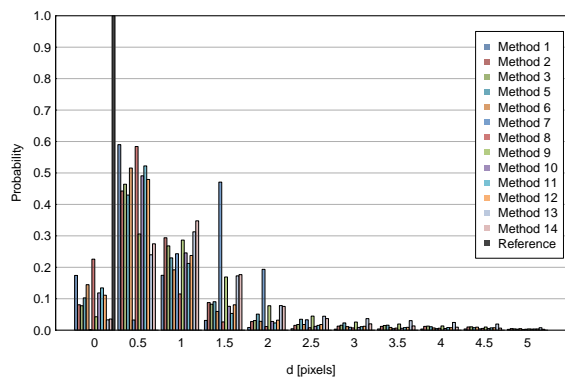


Scenario 1 Low Density

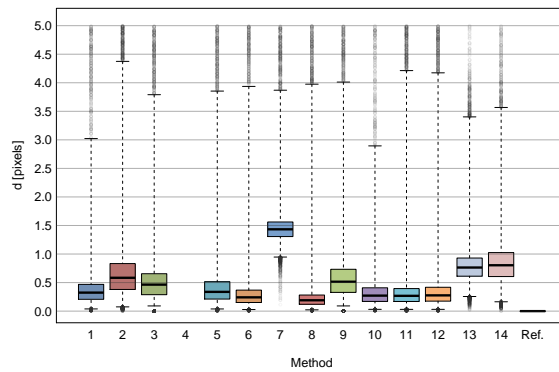
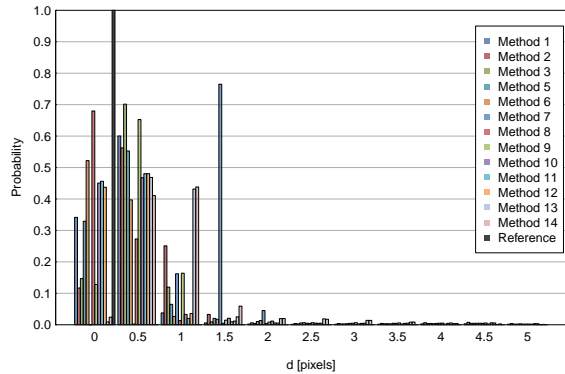
SNR = 1



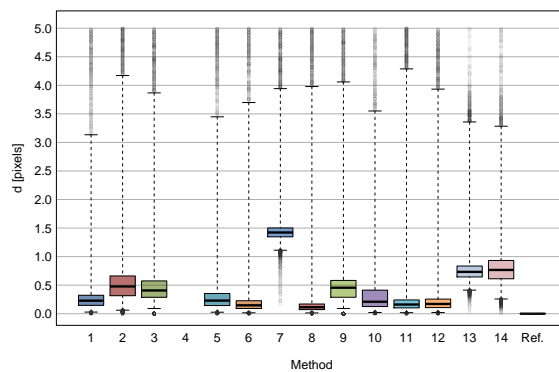
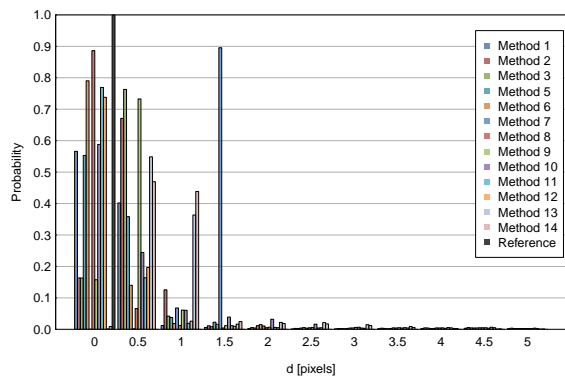
SNR = 2



SNR = 4

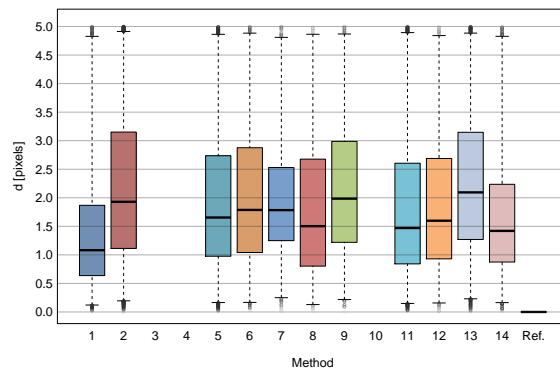
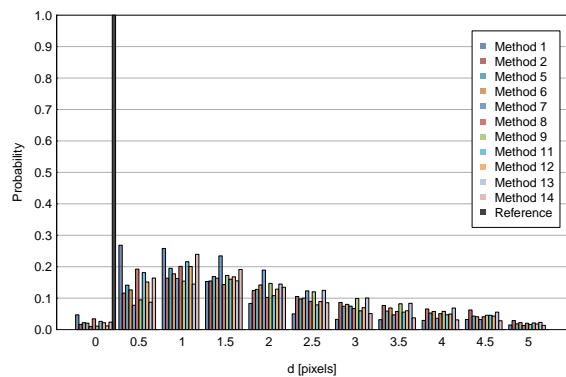


SNR = 7

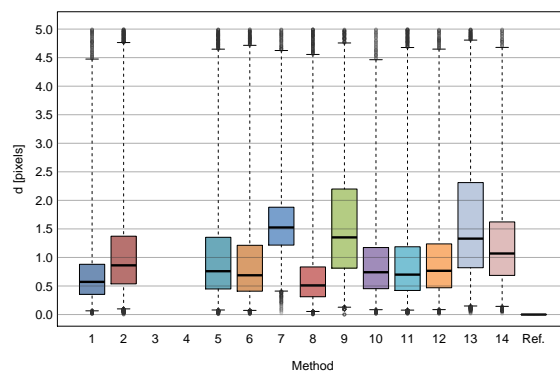
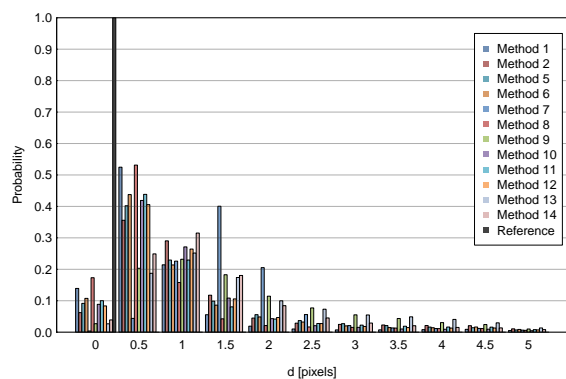


Scenario 1 Mid Density

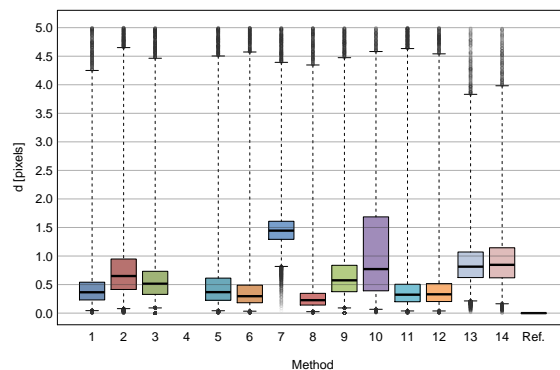
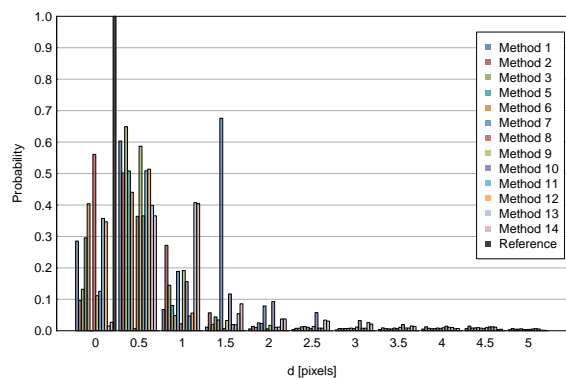
SNR = 1



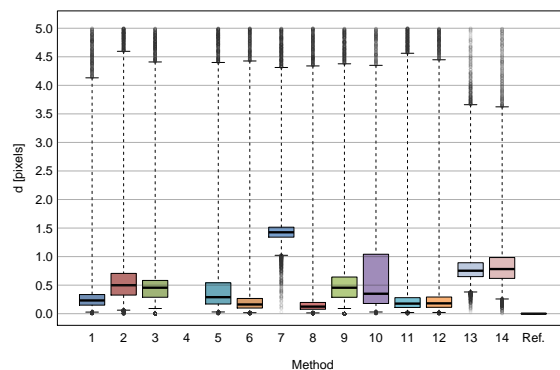
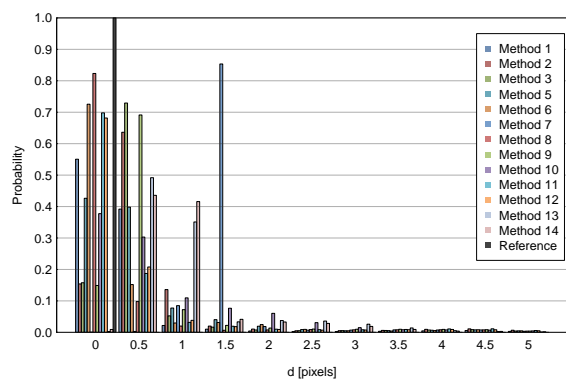
SNR = 2



SNR = 4



SNR = 7

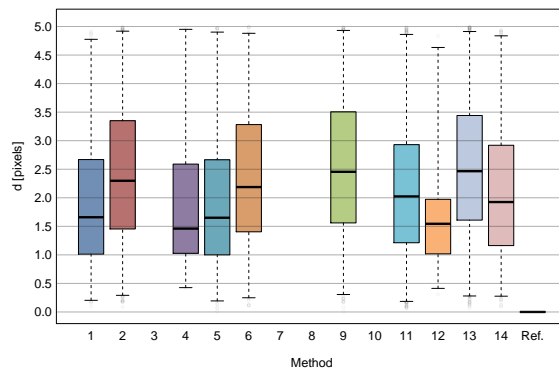
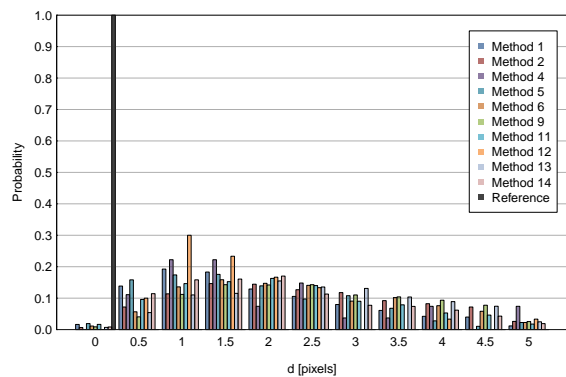


Scenario 1 High Density

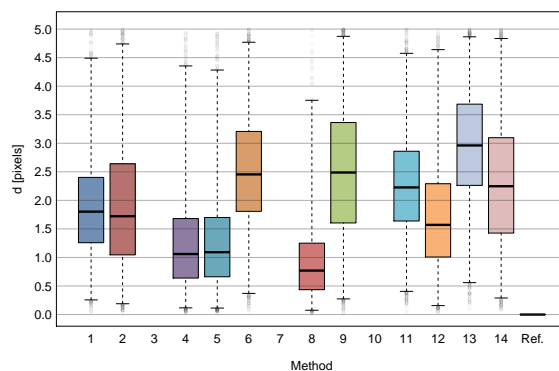
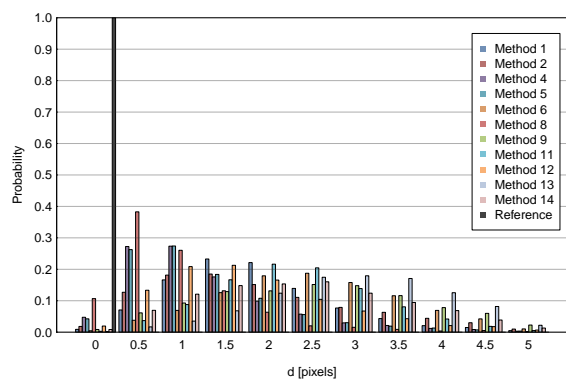
Supplementary Figure 10: Localization Plots for Scenario 2 Data

The following pages show the (normalized) localization histograms (left plots) and corresponding box plots (right plots) for the 12 image sequences of Scenario 2 in the competition data set. Each page shows the results for one density level (Low, Mid, High) and the four different SNR levels (1, 2, 4, 7). Each plot shows the results for all applicable particle tracking methods as well as the ground-truth (reference) data using the same color coding throughout. In the box plots, the boxes indicate the 25th-75th percentile, the horizontal bar inside a box indicates the median value, the whiskers range from the 1st to the 99th percentile, and the gray circles above/below the whiskers indicate the outliers.

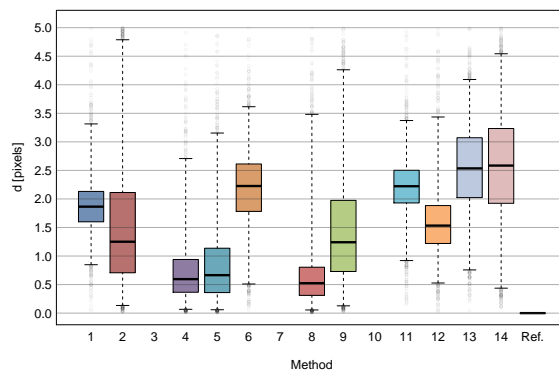
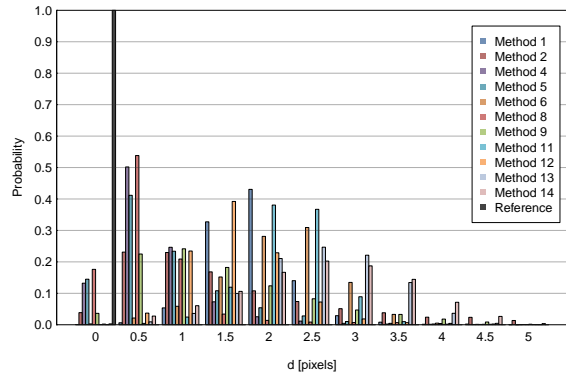
SNR = 1



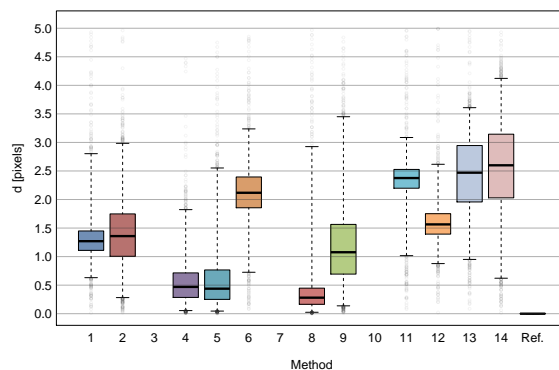
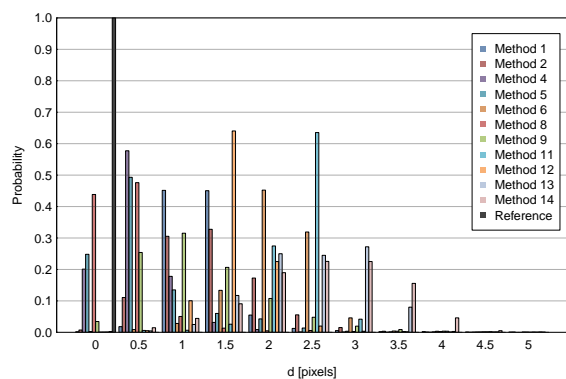
SNR = 2



SNR = 4

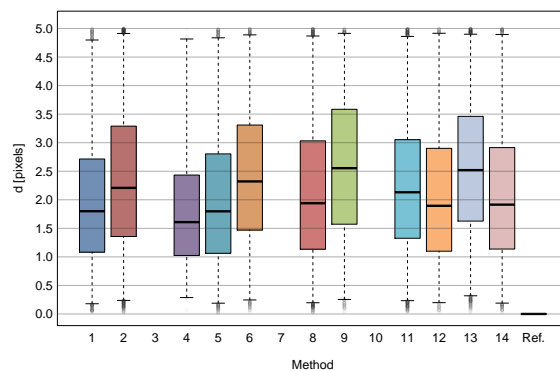
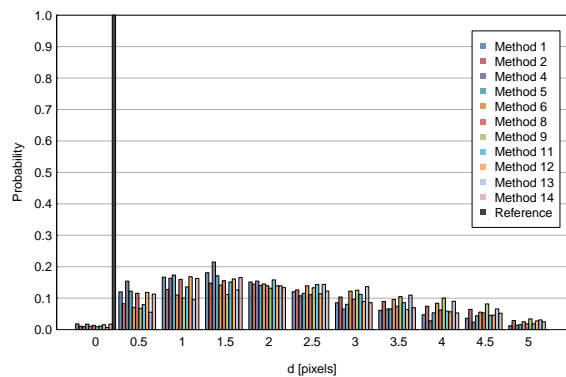


SNR = 7

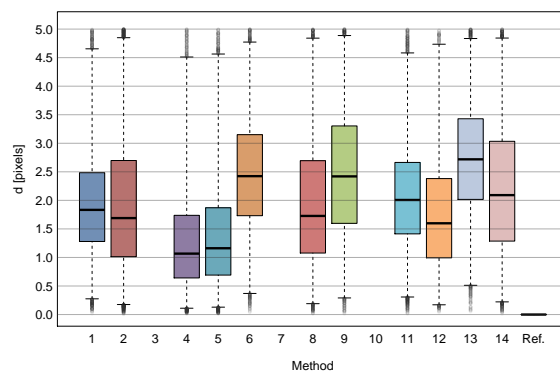
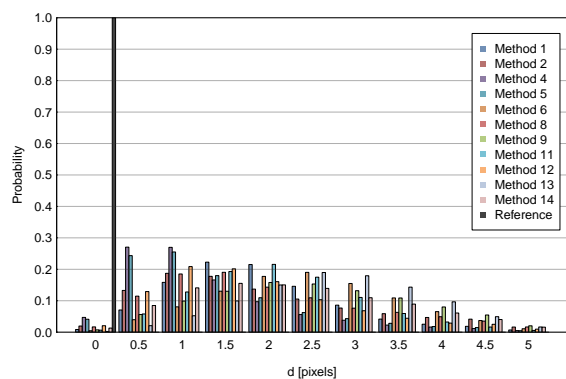


Scenario 2 Low Density

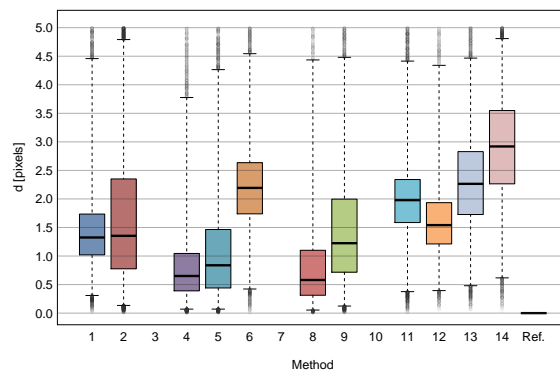
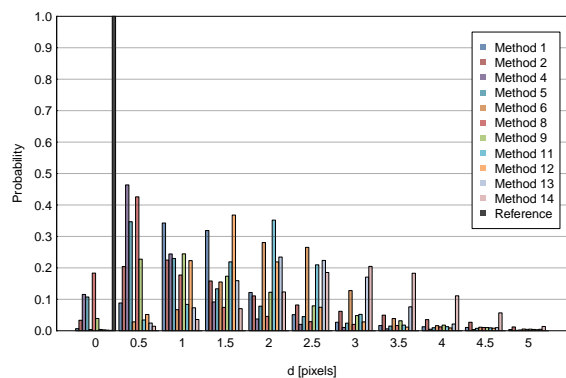
SNR = 1



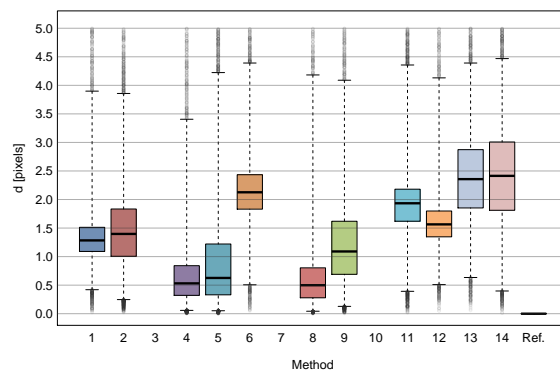
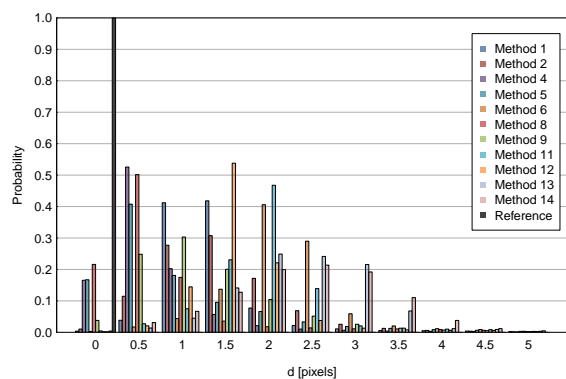
SNR = 2



SNR = 4

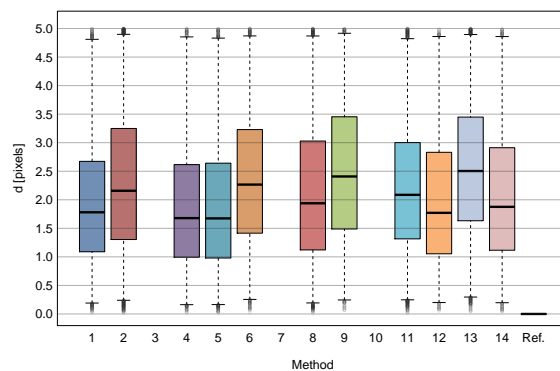
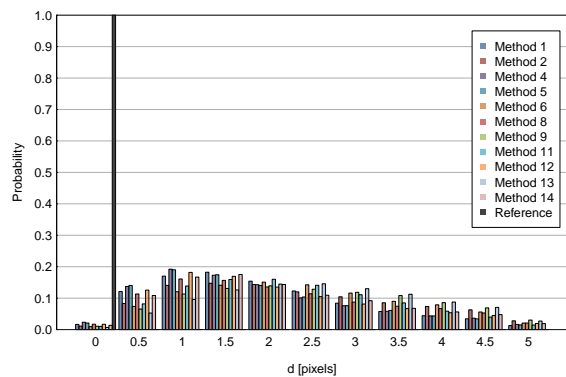


SNR = 7

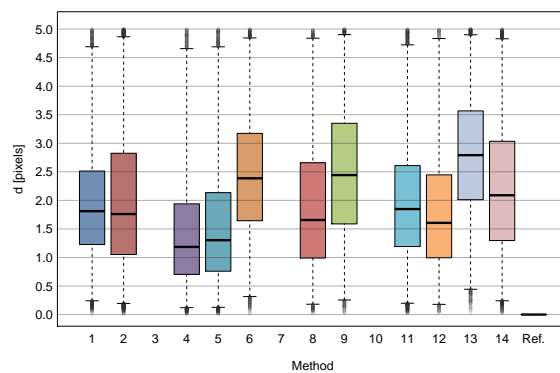
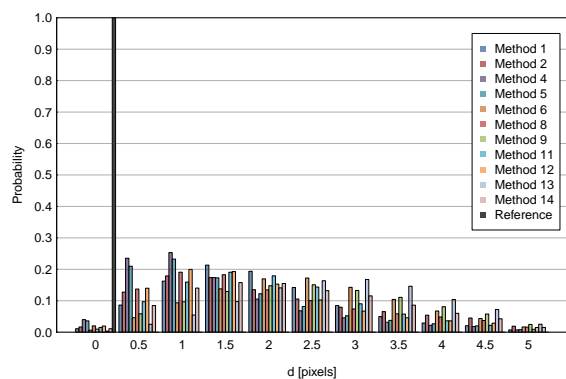


Scenario 2 Mid Density

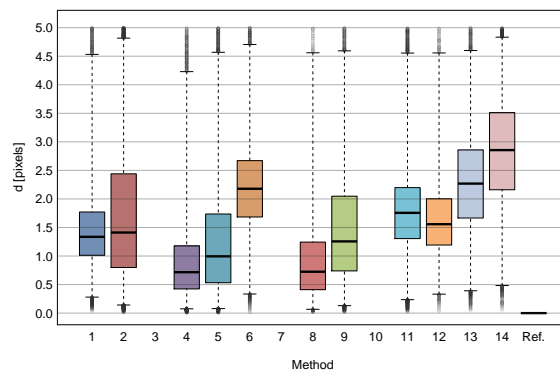
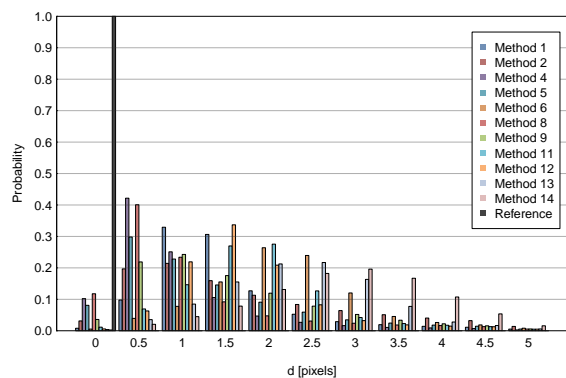
SNR = 1



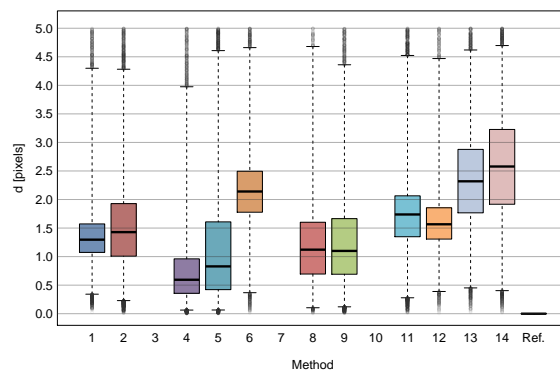
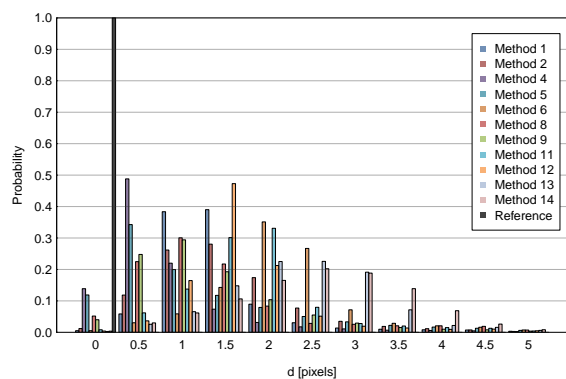
SNR = 2



SNR = 4



SNR = 7

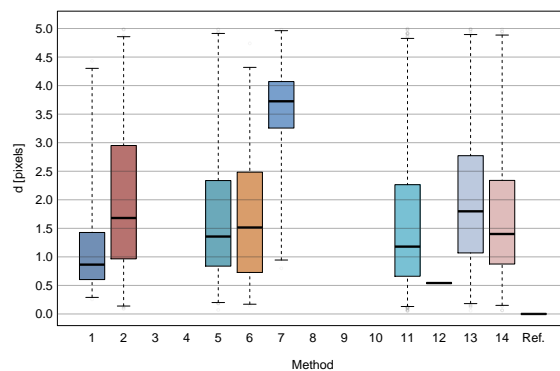
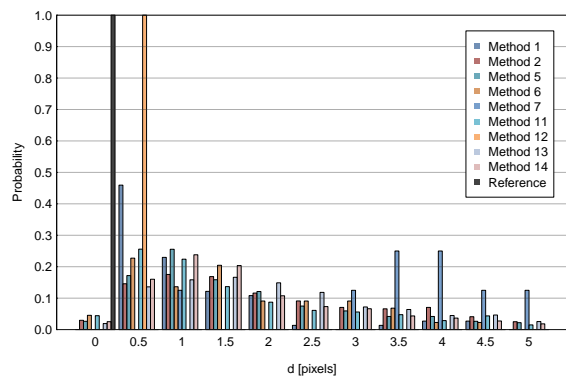


Scenario 2 High Density

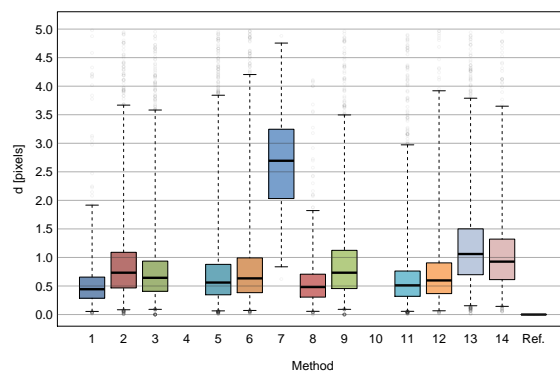
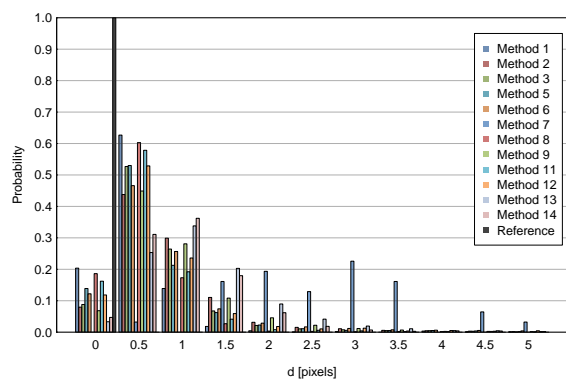
Supplementary Figure 11: Localization Plots for Scenario 3 Data

The following pages show the (normalized) localization histograms (left plots) and corresponding box plots (right plots) for the 12 image sequences of Scenario 3 in the competition data set. Each page shows the results for one density level (Low, Mid, High) and the four different SNR levels (1, 2, 4, 7). Each plot shows the results for all applicable particle tracking methods as well as the ground-truth (reference) data using the same color coding throughout. In the box plots, the boxes indicate the 25th-75th percentile, the horizontal bar inside a box indicates the median value, the whiskers range from the 1st to the 99th percentile, and the gray circles above/below the whiskers indicate the outliers.

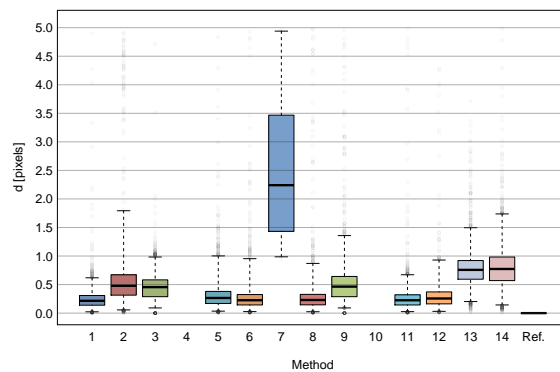
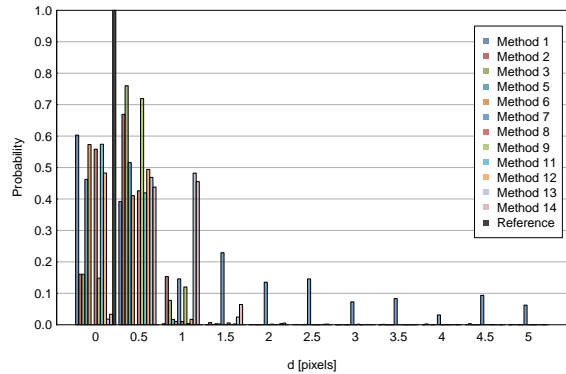
SNR = 1



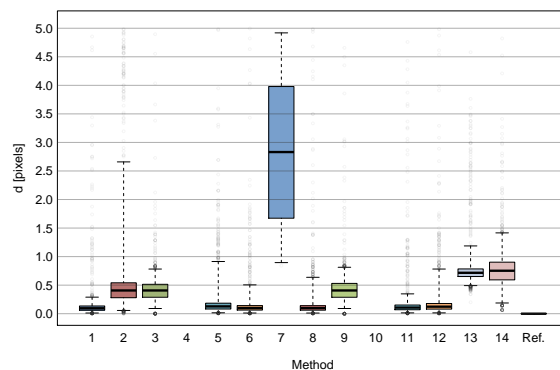
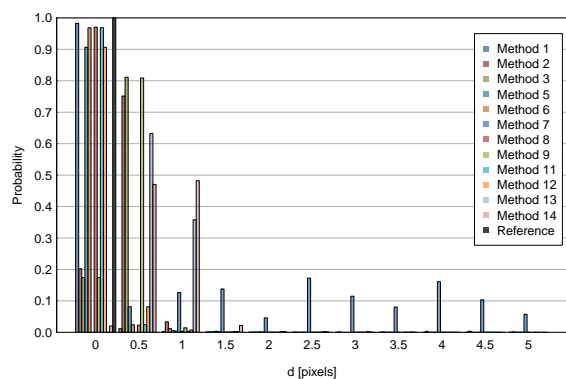
SNR = 2



SNR = 4

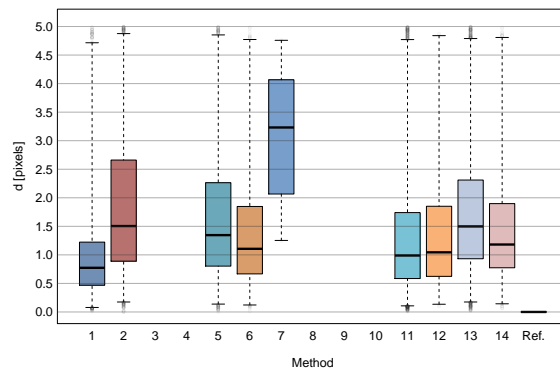
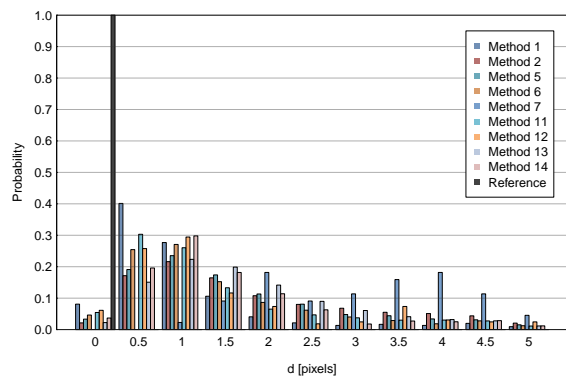


SNR = 7

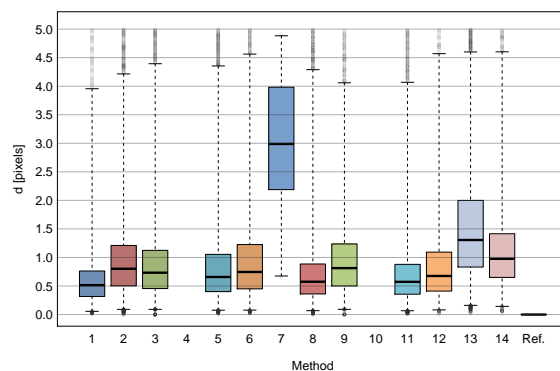
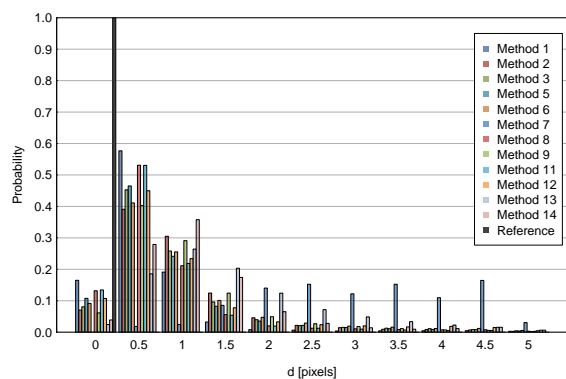


Scenario 3 Low Density

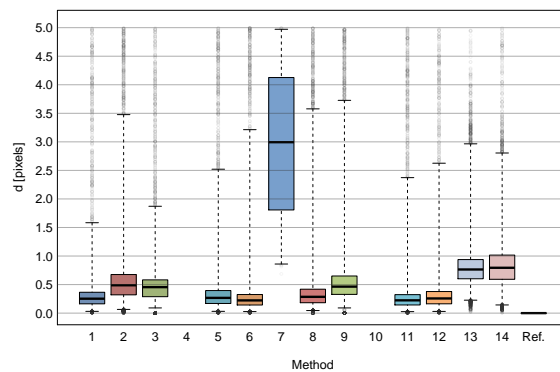
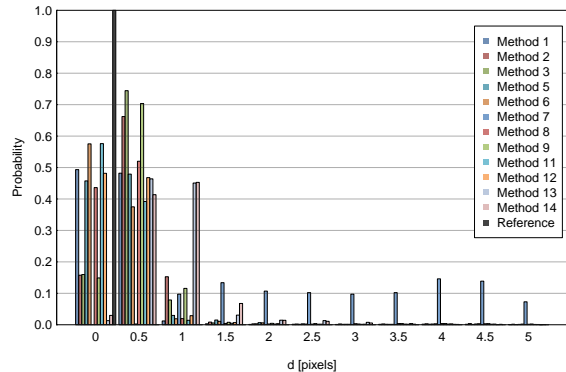
SNR = 1



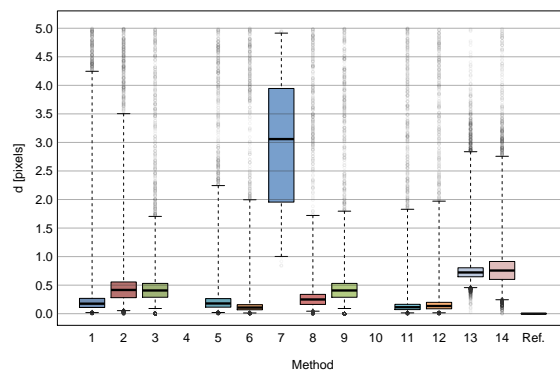
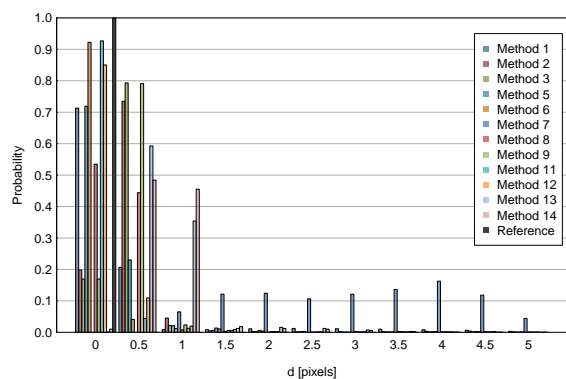
SNR = 2



SNR = 4

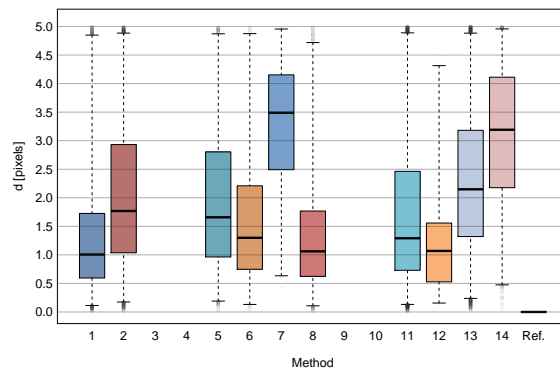
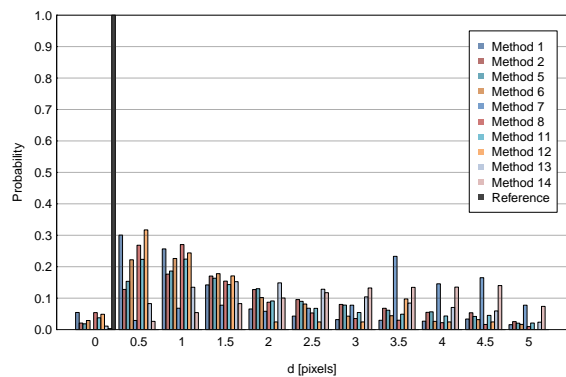


SNR = 7

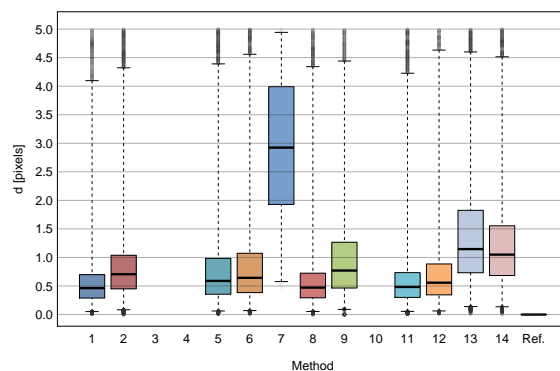
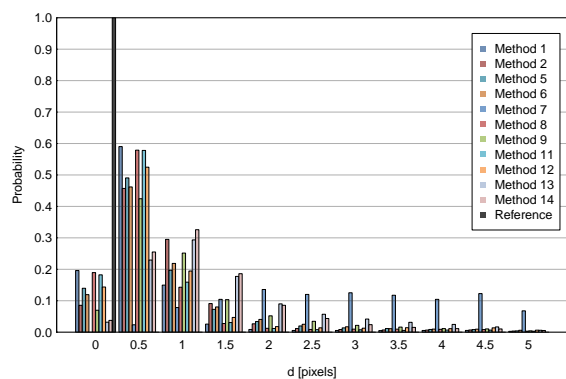


Scenario 3 Mid Density

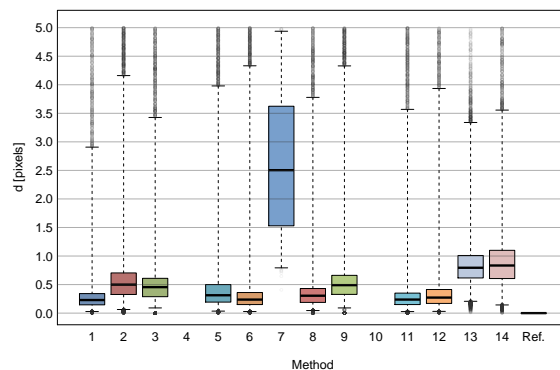
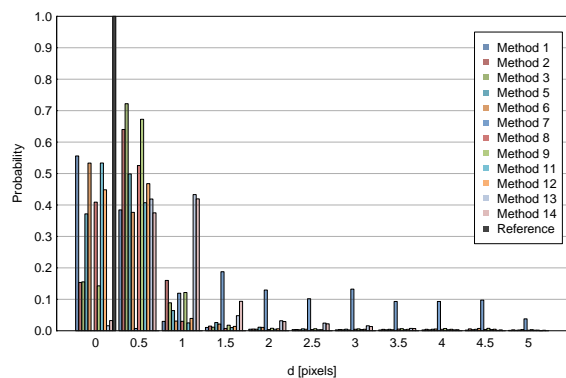
SNR = 1



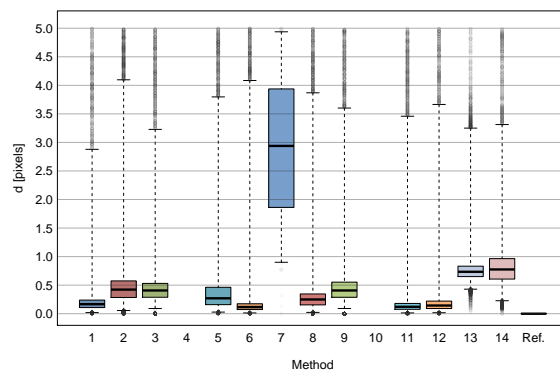
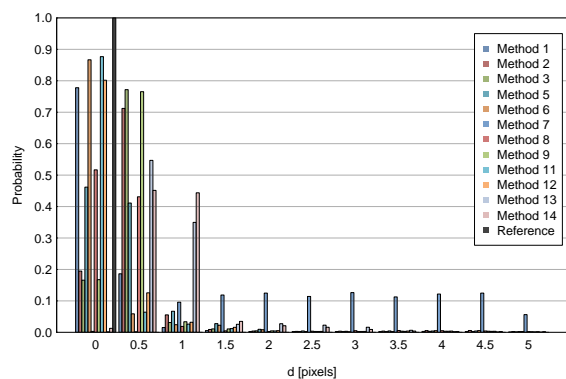
SNR = 2



SNR = 4



SNR = 7

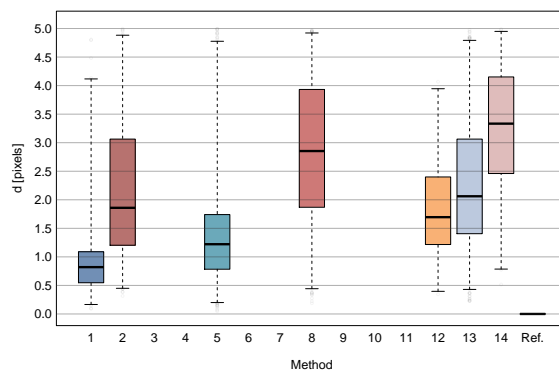
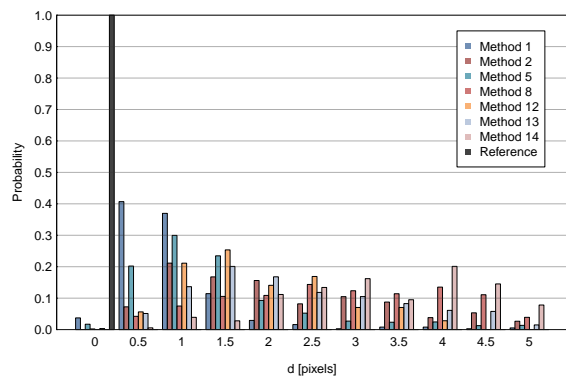


Scenario 3 High Density

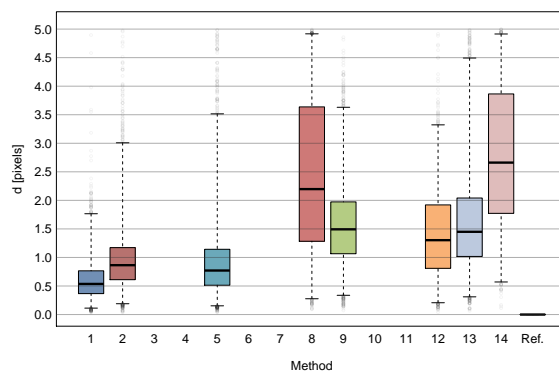
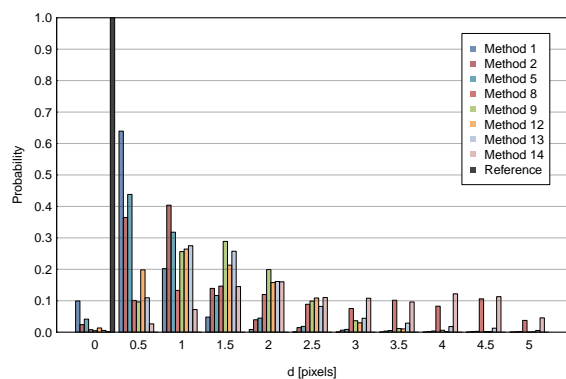
Supplementary Figure 12: Localization Plots for Scenario 4 Data

The following pages show the (normalized) localization histograms (left plots) and corresponding box plots (right plots) for the 12 image sequences of Scenario 4 in the competition data set. Each page shows the results for one density level (Low, Mid, High) and the four different SNR levels (1, 2, 4, 7). Each plot shows the results for all applicable particle tracking methods as well as the ground-truth (reference) data using the same color coding throughout. In the box plots, the boxes indicate the 25th-75th percentile, the horizontal bar inside a box indicates the median value, the whiskers range from the 1st to the 99th percentile, and the gray circles above/below the whiskers indicate the outliers.

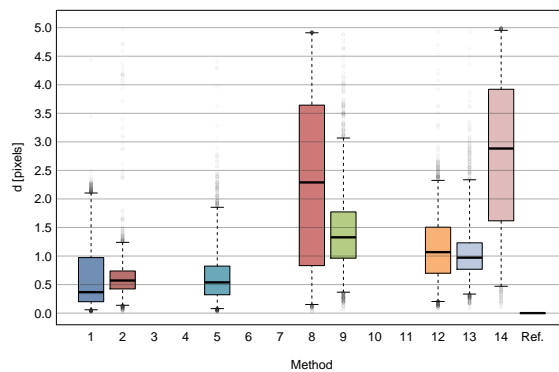
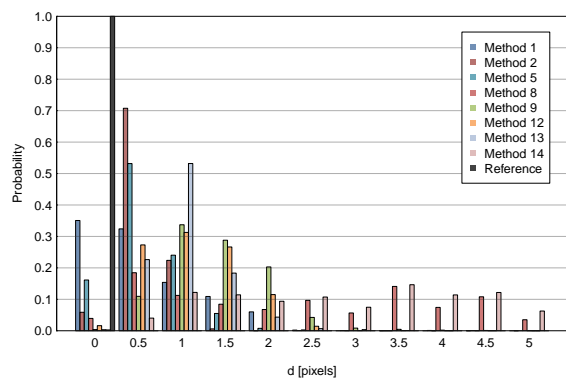
SNR = 1



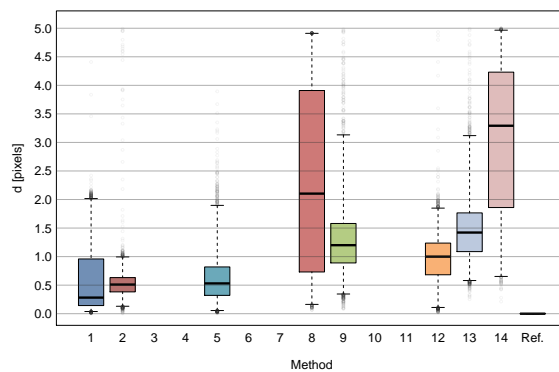
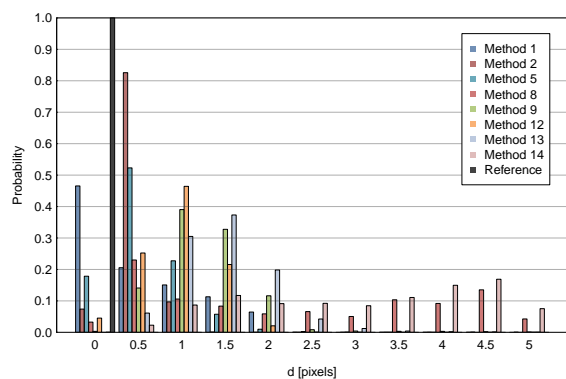
SNR = 2



SNR = 4

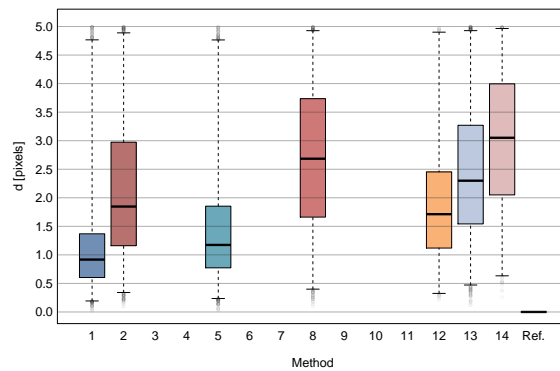
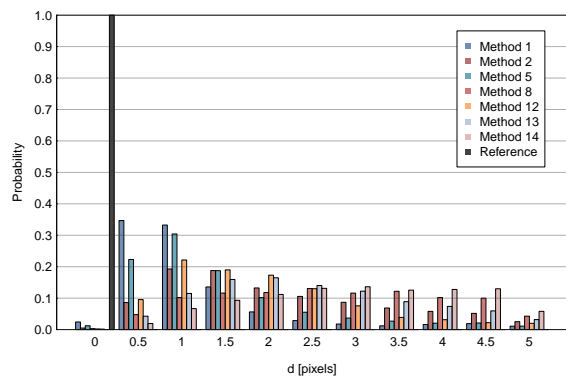


SNR = 7

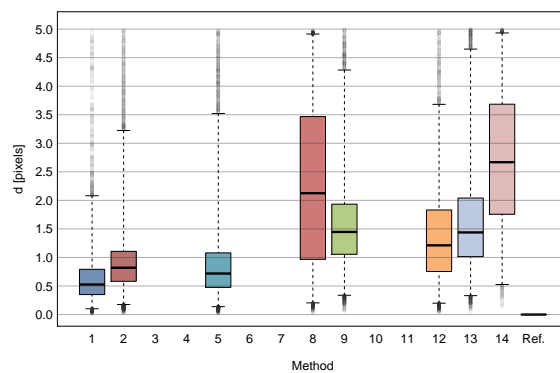
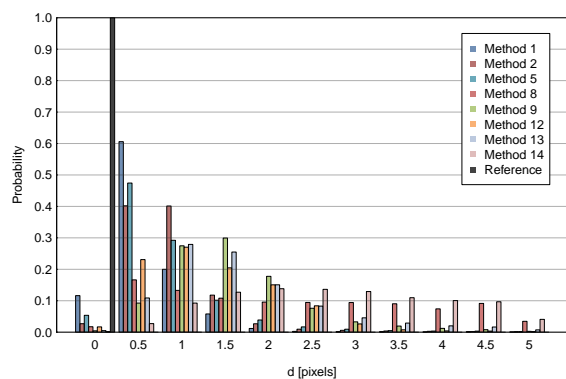


Scenario 4 Low Density

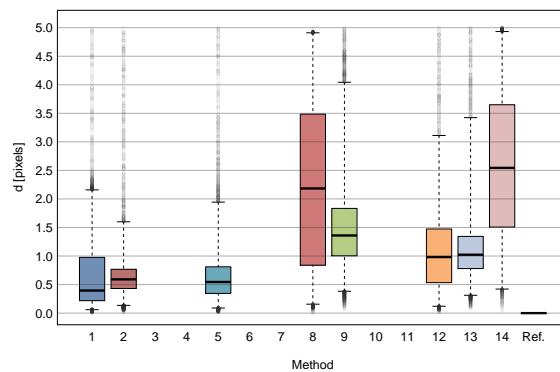
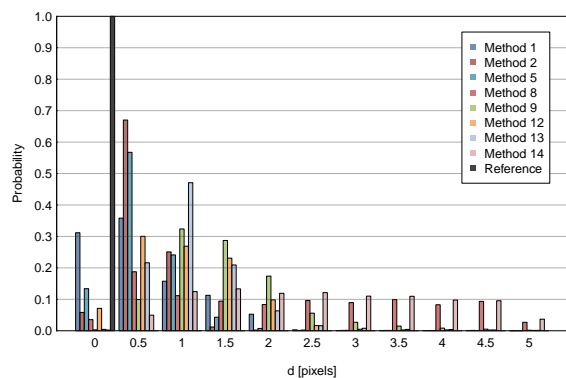
SNR = 1



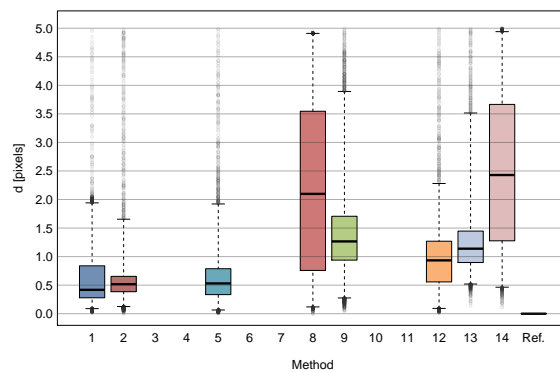
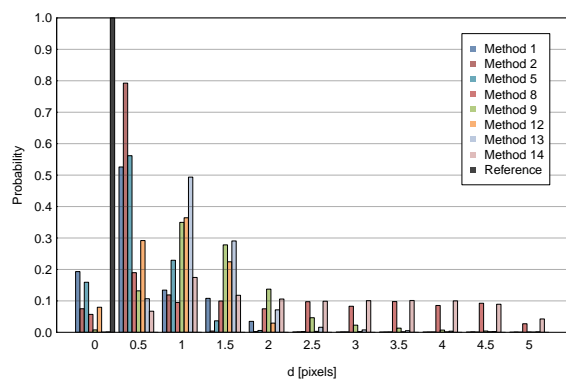
SNR = 2



SNR = 4

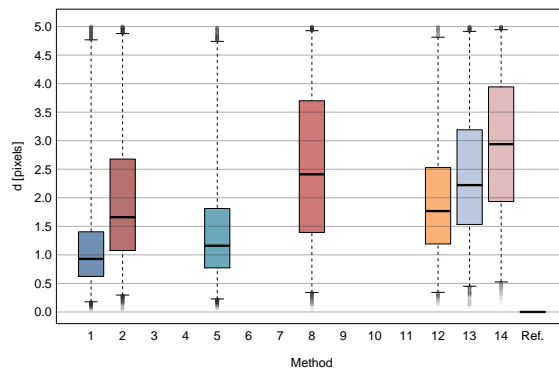
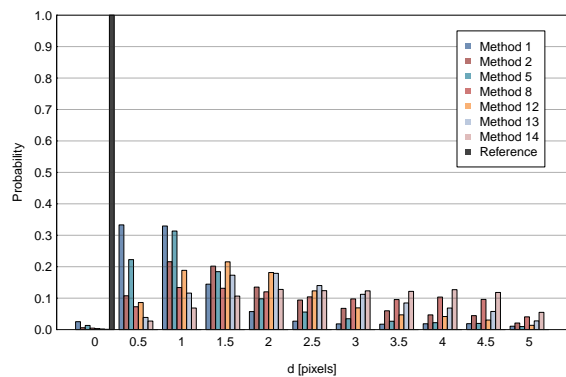


SNR = 7

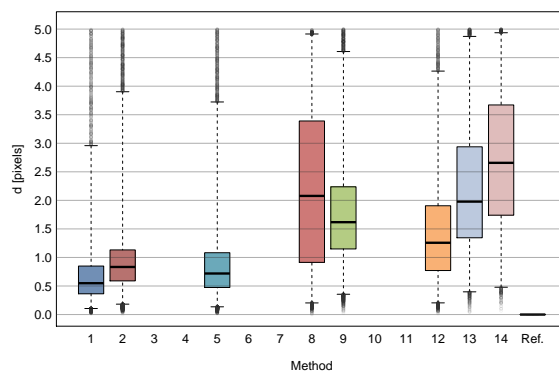
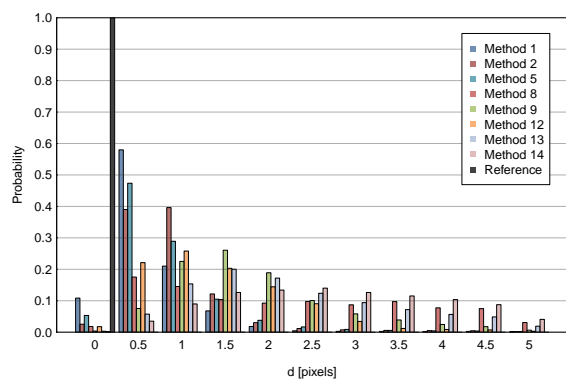


Scenario 4 Mid Density

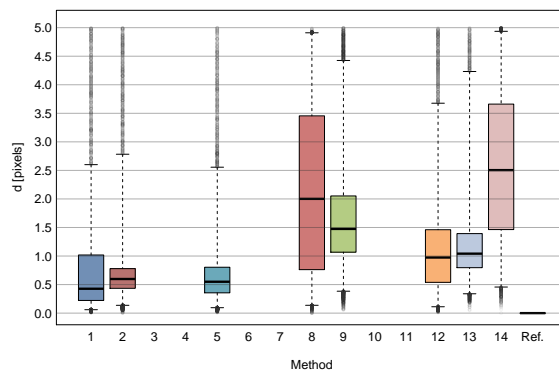
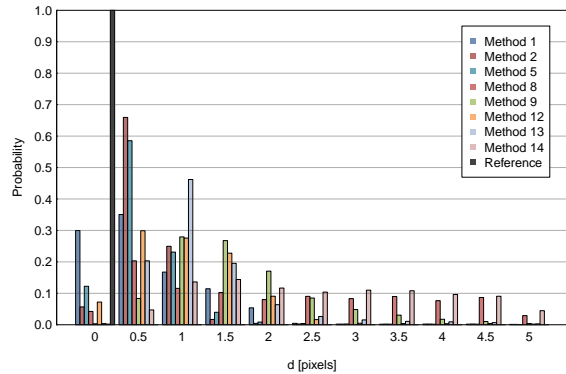
SNR = 1



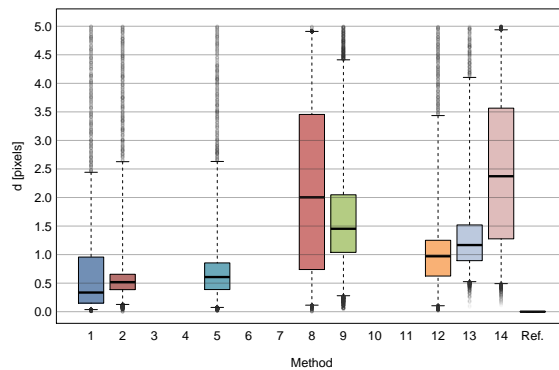
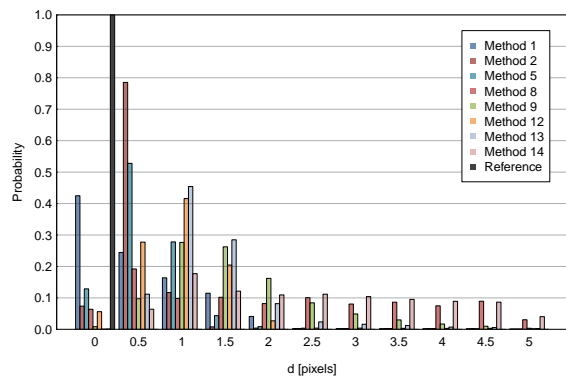
SNR = 2



SNR = 4



SNR = 7



Scenario 4 High Density

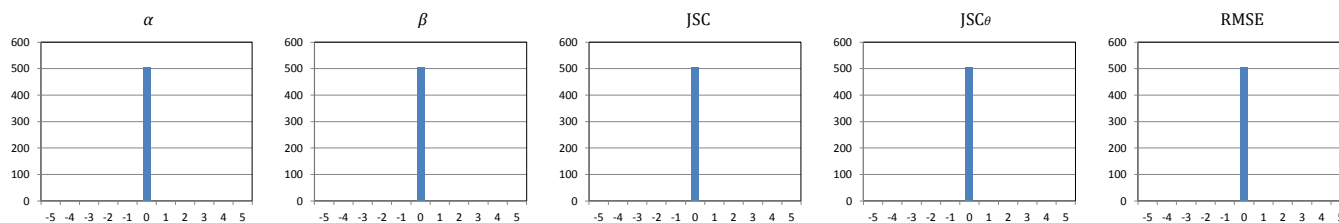
Supplementary Table 5: Top-3 Best Methods for Different Gates

The following sub-tables show the top-3 best performing methods for decreasing values (5, 4, 3, 2, 1 pixel) of the gate parameter ϵ described in **Supplementary Note 3**. Below each sub-table the histogram of ranking differences compared to $\epsilon = 5$ pixels are shown for each performance measure. As described in the main paper, of the 48 (cases) \times 14 (methods) = 672 possible tracking results, 505 were submitted, amounting to the same number of rankings per measure (each method was ranked for each case according to that measure). By changing ϵ , each method could potentially increase or decrease in rank, up to 13 places. Since in most cases the change in the ranking of any method was less than 5 places, only the portion of the histograms within that range are shown.

Top-3 for $\varepsilon = 5$

Scenario	Density SNR	Vesicles												Microtubules												Receptors												Viruses											
		Low				Mid				High				1	2	4	7	1	2	4	7	1	2	4	7	1	2	4	7	1	2	4	7	1	2	4	7												
		1	2	4	7	1	2	4	7	1	2	4	7																									1	2	4	7	1	2	4	7	1	2	4	7
α	#1	5	2	5	1	2	8	1	1	2	8	1	1	11	5	4	4	2	2	4	4	2	2	4	4	11	11	11	11	11	8	3	3	11	11	3	3	5	5	1	1	1	2	2	2	2			
	#2	2	5	11	8	5	2	8	8	5	2	3	5	13	4	8	5	11	4	2	2	11	11	2	2	14	5	1	1	13	11	11	11	13	2	11	11	13	2	5	5	5	5	5	5	5			
	#3	11	11	3	5	11	11	3	3	11	5	5	8	2	2	5	2	8	11	5	5	8	4	11	5	5	2	3	8	1	2	1	5	2	5	1	1	1	8	1	2	2	2	1	1	1	1		
β	#1	5	2	5	1	5	8	8	8	5	8	1	1	13	5	4	4	11	4	4	4	11	11	4	4	13	11	11	11	11	2	3	3	11	11	3	3	5	5	1	1	5	5	2	2	2	2		
	#2	2	5	3	5	2	3	1	1	13	5	3	8	9	4	5	5	8	11	5	2	8	4	11	2	11	2	1	1	13	11	11	11	13	2	11	11	8	2	5	5	1	2	2	1	1	2	5	
	#3	13	9	11	8	13	2	3	3	11	11	8	5	2	2	11	8	2	13	5	11	5	1	5	5	11	5	9	3	5	1	5	1	5	1	5	1	1	1	1	1	1	1	1	1	1	1		
JSC	#1	5	2	5	7	5	8	7	7	5	5	7	7	13	5	5	2	11	11	4	2	11	11	4	2	13	11	3	3	11	2	3	3	11	11	3	3	5	5	1	1	5	2	2	2	2			
	#2	2	5	7	3	2	3	3	3	13	8	3	1	9	11	11	4	13	4	11	4	13	1	11	4	11	2	11	11	13	11	11	11	13	2	11	11	8	2	5	5	1	2	2	2	1	5	5	5
	#3	13	13	3	5	13	2	1	1	2	2	1	5	2	2	13	4	5	8	5	5	11	8	4	5	11	5	3	1	1	5	5	1	9	1	5	1	1	13	9	2	2	8	1	1	5	8	1	1
JSC θ	#1	13	13	7	7	13	8	7	7	13	5	7	7	13	9	5	2	13	11	4	2	1	11	4	2	13	2	3	3	13	2	3	3	13	11	3	3	5	2	5	1	5	5	2	2	2	2		
	#2	2	2	5	12	5	3	3	3	5	8	3	5	6	13	11	4	11	5	5	4	11	1	11	4	5	13	1	5	1	5	11	11	1	5	11	11	8	5	1	5	13	2	2	5	13	2	5	5
	#3	7	5	3	3	2	5	8	8	14	11	5	1	2	2	5	8	5	8	13	11	5	8	4	5	11	14	9	11	11	11	9	1	8	11	2	1	1	1	1	9	2	2	1	1	1	2	2	1
RMSE	#1	12	1	8	1	1	1	1	1	12	1	1	1	12	8	8	8	4	4	4	4	5	4	4	4	12	1	1	1	12	1	1	11	12	1	1	1	1	1	1	2	2	1	1	2	2	2	2	
	#2	1	12	10	5	12	8	10	12	1	8	8	12	5	4	4	4	1	5	8	8	4	5	8	9	1	12	11	6	1	12	11	12	8	11	11	11	5	2	5	5	5	5	5	5	5			
	#3	6	10	5	6	14	12	8	6	14	12	12	8	1	5	5	5	12	12	5	5	1	12	5	5	6	8	8	6	11	11	11	12	6	1	8	12	12	12	5	1	1	12	2	1	1	1	1	

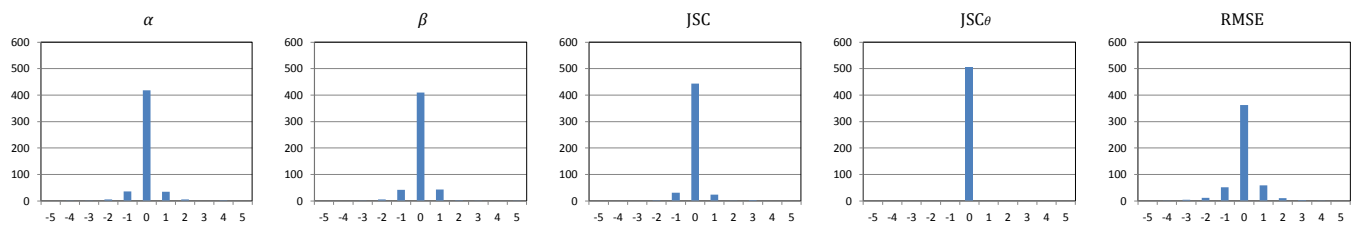
Histograms of ranking differences compared to $\varepsilon = 5$



Top-3 for $\varepsilon = 4$

Scenario	Density SNR	Vesicles												Microtubules												Receptors												Viruses												
		Low				Mid				High				1	2	4	7	1	2	4	7	1	2	4	7	1	2	4	7	1	2	4	7	1	2	4	7													
		1	2	4	7	1	2	4	7	1	2	4	7																																					
α	#1	5	2	5	1	2	8	8	1	2	8	1	1	11	5	4	4	2	2	4	4	2	2	4	4	11	11	11	11	11	8	3	11	11	11	3	3	5	5	1	1	1	2	5	2	1	2	2	2	
	#2	2	5	11	8	5	2	1	8	5	5	3	8	13	4	8	5	8	4	5	2	11	11	2	2	13	5	1	1	13	11	11	3	13	2	11	11	13	2	5	5	5	5	5	1	5	5	5	1	
	#3	11	11	3	5	11	11	3	5	11	2	8	5	2	2	2	5	8	11	11	2	5	8	4	5	5	5	2	5	8	1	2	1	5	1	5	1	1	1	1	1	1	1	1	1	1	1	1	1	1
β	#1	5	2	5	1	5	8	8	8	5	8	1	1	13	5	4	4	11	4	4	4	11	11	4	4	13	11	11	11	11	11	3	11	11	11	3	3	5	5	1	1	5	5	2	2	5	5	2	2	
	#2	2	5	3	5	2	3	1	1	11	5	8	8	5	4	8	5	8	5	2	8	4	5	2	11	2	1	1	13	2	11	3	1	2	11	11	1	2	5	5	1	1	2	5	5	1	1	2	5	5
	#3	13	9	11	8	13	2	3	3	13	11	3	5	9	11	5	8	2	11	11	5	1	5	11	5	5	9	3	5	1	5	1	5	13	5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
JSC	#1	5	2	5	7	5	8	7	7	5	5	7	7	13	5	5	2	11	11	4	2	11	11	4	2	13	11	3	3	11	2	3	3	11	11	3	3	5	5	1	1	5	2	2	2	5	2	2	2	
	#2	2	5	7	5	2	3	3	3	13	8	3	1	9	11	11	4	13	4	11	4	8	4	11	4	11	2	11	11	13	11	11	11	13	2	11	11	8	2	5	5	1	2	2	2	1	5	5	5	5
	#3	13	13	3	3	13	2	1	1	2	2	1	5	2	6	4	5	8	5	5	11	13	1	5	11	5	5	3	1	1	5	5	1	9	1	5	1	1	1	1	1	1	1	1	1	1	1	1	1	1
JSC θ	#1	13	13	7	7	13	8	7	7	13	5	7	7	13	9	5	2	13	11	4	2	1	11	4	2	13	2	3	3	13	2	3	3	13	11	3	3	5	2	5	1	5	5	2	2	5	2	2	2	
	#2	2	2	5	12	5	3	3	3	5	8	3	5	6	13	11	4	11	5	5	4	11	1	11	4	5	13	1	5	1	5	11	11	1	5	11	11	8	5	1	5	13	2	2	5	13	2	5	5	
	#3	7	5	3	3	2	5	8	8	14	11	5	1	2	5	8	5	8	13	11	5	8	4	5	11	14	9	11	11	11	9	1	8	11	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
RMSE	#1	1	1	8	1	1	1	1	1	1	1	8	1	12	8	8	8	4	4	4	4	4	4	4	4	12	1	1	6	12	1	1	11	12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	#2	12	12	11	12	12	8	8	12	12	8	1	12	1	4	4	4	12	5	8	8	5	5	8	5	1	8	11	1	1	12	11	6	1	11	11	11	5	5	5	5	5	5	5	5	5	5	5	5	
	#3	11	10	6	8	14	12	12	8	14	12	12	8	5	5	5	5	1	12	5	5	12	12	5	8	11	12	6	8	11	11	12	12	8	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12

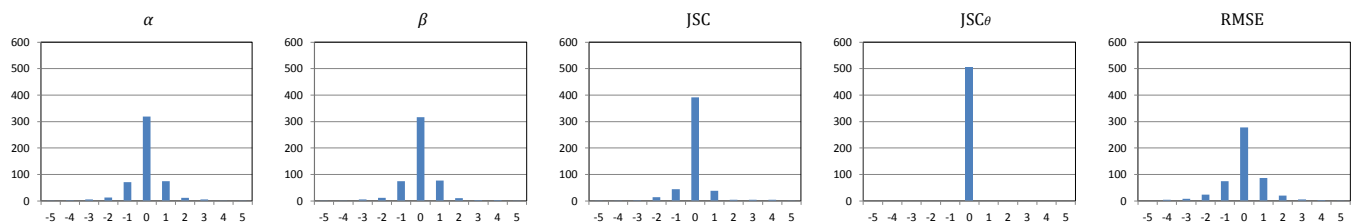
Histograms of ranking differences compared to $\varepsilon = 5$



Top-3 for $\varepsilon = 3$

Scenario	Density SNR	Vesicles												Microtubules												Receptors												Viruses												
		Low				Mid				High				Low				Mid				High				Low				Mid				High				Low				Mid				High				
		1	2	4	7	1	2	4	7	1	2	4	7	1	2	4	7	1	2	4	7	1	2	4	7	1	2	4	7	1	2	4	7	1	2	4	7	1	2	4	7	1	2	4	7					
α	#1	5	2	5	1	2	8	8	1	2	8	1	1	11	5	4	4	8	4	4	4	11	2	4	4	11	11	11	11	11	8	3	11	11	11	3	11	5	1	1	1	2	5	2	1	2	2	2		
	#2	2	5	11	8	5	2	1	8	5	5	8	8	13	4	8	5	11	2	5	5	2	4	2	2	14	5	1	1	13	11	11	3	1	2	11	3	13	2	5	2	5	5	2	1	5	5	5	1	
	#3	11	11	8	5	11	11	5	5	11	2	3	5	2	2	5	8	2	5	2	2	8	11	5	5	5	2	5	2	5	8	1	2	1	5	13	1	1	1	1	1	2	5	2	1	1	5	1		
β	#1	5	2	5	1	5	8	8	8	5	8	1	1	13	5	4	4	11	4	4	4	11	4	4	4	11	11	11	11	11	3	11	11	11	3	3	5	5	1	1	5	5	5	2	5	5	2	2		
	#2	2	5	11	8	2	11	1	1	11	5	8	8	5	4	8	5	8	5	5	2	8	11	5	2	13	2	1	1	13	2	11	3	1	2	11	11	1	2	5	5	1	2	2	1	1	5	1		
	#3	13	9	3	5	13	3	3	5	13	11	3	5	2	8	5	8	1	11	1	5	1	5	11	5	5	9	5	5	1	8	1	5	13	5	1	1	1	1	2	2	8	1	1	5	8	2	1	5	
JSC	#1	5	2	5	7	5	8	7	7	5	5	7	7	13	5	5	2	11	11	4	2	11	11	4	2	13	11	3	3	11	2	3	3	11	11	3	3	5	5	1	1	5	2	2	2	2	2			
	#2	2	5	7	5	2	3	3	3	13	8	1	1	2	4	4	4	8	4	11	4	8	4	11	4	11	2	11	11	13	11	11	11	13	2	11	11	8	2	5	5	1	5	5	5	5				
	#3	13	13	3	3	13	2	1	1	2	2	3	5	9	11	11	5	13	5	5	11	1	1	5	11	5	5	3	1	1	1	5	1	9	1	5	1	1	1	1	9	2	2	8	1	1	2	2	1	1
JSC θ	#1	13	13	7	7	13	8	7	7	13	5	7	7	13	9	5	2	13	11	4	2	1	11	4	2	13	2	3	3	13	2	3	3	13	11	3	3	5	2	5	1	5	5	2	2	2	2			
	#2	2	2	5	12	5	3	3	3	5	8	3	5	6	13	11	4	11	5	5	4	11	1	11	4	5	13	1	5	1	5	11	11	1	5	11	11	8	5	1	5	13	2	5	5	13	2	5	5	
	#3	7	5	3	3	2	5	8	8	14	11	5	1	2	5	8	5	8	13	11	5	8	4	5	11	14	9	11	11	11	9	1	8	11	2	1	1	1	1	9	2	2	1	1	1	2	2	1	1	1
RMSE	#1	1	1	8	1	1	8	8	1	1	1	8	1	12	8	8	8	4	4	4	8	4	4	4	4	12	1	1	8	1	1	11	11	12	1	11	1	1	1	1	2	2	1	1	2	2	2	2		
	#2	12	12	11	8	12	1	12	8	12	8	12	12	5	4	4	4	5	5	8	4	5	5	8	5	1	8	11	6	12	12	1	6	1	8	1	11	5	5	5	5	5	5	5	5	5	5			
	#3	11	6	6	11	11	12	1	12	11	12	1	8	1	5	5	5	12	12	5	5	12	12	5	8	11	12	8	11	11	11	12	12	8	11	8	12	2	2	1	1	12	2	1	1	2	2	1	1	

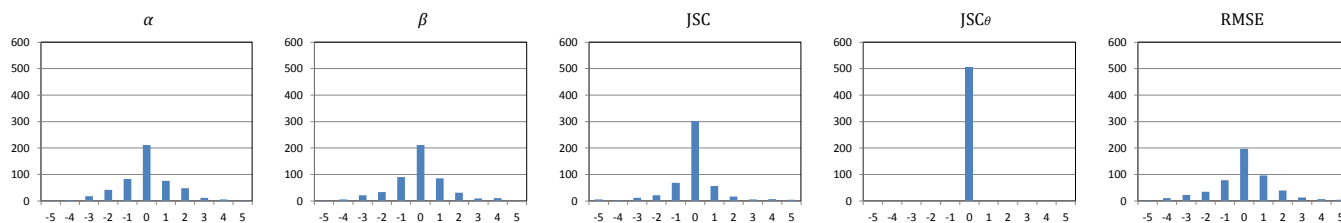
Histograms of ranking differences compared to $\varepsilon = 5$



Top-3 for $\varepsilon = 2$

Scenario	Density SNR	Vesicles												Microtubules												Receptors												Viruses																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
		Low				Mid				High				Low				Mid				High				Low				Mid				High																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
		1	2	4	7	1	2	4	7	1	2	4	7	1	2	4	7	1	2	4	7	1	2	4	7	1	2	4	7	1	2	4	7	1	2	4	7	1	2	4	7																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
α	#1	5	5	5	1	5	8	8	8	11	8	1	1	11	5	8	4	8	4	4	11	4	4	4	11	11	11	1	11	8	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11</

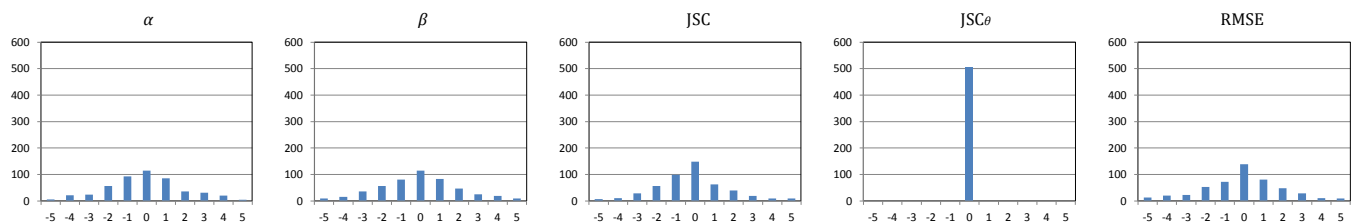
Histograms of ranking differences compared to $\varepsilon = 5$



Top-3 for $\varepsilon = 1$

Scenario	Density SNR	Vesicles												Microtubules												Receptors												Viruses																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
		Low				Mid				High				Low				Mid				High				Low				Mid				High				Low				Mid				High																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
		1	2	4	7	1	2	4	7	1	2	4	7	1	2	4	7	1	2	4	7	1	2	4	7	1	2	4	7	1	2	4	7	1	2	4	7	1	2	4	7	1	2	4	7																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
α	#1	5	5	8	8	11	8	8	8	1	8	8	1	11	5	8	8	8	4	4	4	8	4	4	4	11	11	1	1	11	8	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11

Histograms of ranking differences compared to $\varepsilon = 5$



Supplementary Note 1: Particle Tracking Methods

This document provides a description of all particle tracking methods evaluated in this study. In total 14 teams participated with their own, custom developed methods. Generally speaking, particle tracking methods consist of two steps: 1) particle detection (the spatial aspect), in which spots that stand out sufficiently from the background according to certain criteria are identified in every frame of the image sequence, and 2) particle linking (the temporal aspect), in which detected particles are connected from frame to frame using another set of criteria, to form tracks.¹⁻⁶ Here we explain the concepts underlying the detection and the linking step of each method. A summary of the methods is given in **Table 1** of the main paper. We also summarize the user parameters of each method and typical values. The exact parameter values for each method and each image sequence in this study are given in **Supplementary Table 1**. Software implementations of the methods are either publicly available or can be requested from the corresponding authors, as indicated for each method.

METHOD 1

Authors: Ivo F. Sbalzarini, Yuanhao Gong, Janick Cardinale

Email: ivos@mpi-cbg.de

Software: <http://mosaic.mpi-cbg.de/?q=downloads>

Form: ImageJ plugin

METHOD 1: INTRODUCTION

This method was designed for applications where large numbers of images of low signal-to-noise ratio (SNR) need to be analyzed in the absence of prior knowledge about the type of motion. The former is typically the case in high-throughput screens, while the latter is a common situation during the explorative phase of discovery. There, one does not wish to assume any motion model or prior, in order not to bias the results toward the expected type of motion, that is, not to miss or overlook unexpected motion patterns in the data. The design goals of the method hence were computational efficiency, prior-freeness, ease of use, and robust detection performance at low SNR. Motion priors can be included into the linking cost function of the algorithm, but are not present in its basic version. Balancing prior information and data fidelity is a fundamental challenge for any particle tracking algorithm, and shortcomings in one can often be compensated by additional effort in the other. Full details about the method can be found in a previous publication.⁷

METHOD 1: DETECTION

Detection is based on finding local maxima at the pixel level followed by iterative intensity-weighted centroid calculation at subpixel resolution. A pixel is taken as a local maximum if no other pixel within a radius of w around it is brighter, and if its intensity is in the upper p -th percentile of all intensity values in the image. Local maxima detection is done by grayscale dilation followed by selecting all pixels that have the same value before and after dilation. Percentiles are computed per frame in order to be robust against possible drift in image intensity over time (for example due to photo-bleaching). Detected candidate points can optionally be filtered in order to reduce false-positive detections. False positives are identified as those points whose intensity moments of order 0 and 2 are significantly different from the moments of the majority of points. This is done by clustering the points in intensity-moment space and discarding all points that are in regions with a density smaller than a given cutoff.⁷ For each retained candidate, the subpixel particle centroid position is estimated as the intensity-weighted centroid within a radius w , updating the pixel-level position if the centroid in any direction is more than half a pixel away, and repeating the centroid calculation until convergence. The radius parameter w should be chosen slightly larger than the apparent radius of a particle in the image, but smaller than the smallest inter-particle distance, which is usually in the range of 2 to 6 pixels for typical images. The percentile parameter p should correspond to the percentage of pixels in the image covered by particles, which is usually in the range of 0.1% to 5%.

METHOD 1: LINKING

Linking of detected particles across frames is done by approximately solving a combinatorial optimization problem under the topological constraint that particles do not split or fuse. Particles entering or leaving the scene are accounted for by linking them to a "dummy". Optimization is done using greedy hill-climbing starting from a nearest-neighbor initialization. The distance metric considers the location, intensity, and the second moment of the intensity distribution of the particles. Infeasible links, determined by a link-length cutoff L corresponding to the largest distance any particle may travel from one frame to the next, are flagged and excluded from the optimization process in order to increase computational efficiency. The optimizer stops when no more cost-improving link can be found, indicating a local optimum. The linking optimization can optionally be done over multiple frames in order to account for intermittent detections (fluorophore blinking) and to close gaps in tracks. This is done by applying the approximate combinatorial optimizer over different frame shifts $r = 1 \dots R$ and choosing the best result. The value of the link-length cutoff parameter L should be set depending on the time

resolution of the image sequence and the velocity of the particles. For well-sampled sequences, typical cutoffs are about 5 to 20 pixels. The link-range parameter R should be set to the number of frames that particles are expected to be missing due to fluorophore blinking. Higher link ranges increase the computational cost and the probability of erroneous links. In practical applications, suitable values are in the range of 1 to 5 (mostly 1 or 2). In the present study, R was fixed to 1, as particles did not blink.

METHOD 1: PARAMETERS

The user parameters of this method (and typical values) are:

- **Radius** for local maxima detection and centroid computation (2-6 pixels).
- **Cutoff** for discarding detections based on intensity moments (0-10).
- **Percentile** of image intensities that detected maxima must be in (0.1%-5%).
- **Link-range** used by the combinatorial optimizer for linking detections (1-5 frames).
- **Link-length** cutoff for linking detections in successive frames (5-20 pixels).

The exact parameter values for each image sequence in this study are given in **Supplementary Table 1**.

METHOD 2

Authors: Craig Carthel and Stefano Coraluppi

Email: stefano.coraluppi@compunetix.com

Software: Contact the authors

Form: Windows executable

METHOD 2: INTRODUCTION

This method was designed with the goal to avoid, to the extent possible, redundant detections per particle. Most automatic multi-object tracking schemes allow for some missed detections but are based on the underlying assumption that each particle gives rise to at most one detection per sensor scan. Repeated multiple detections will invariably give rise to spurious tracks. The detection method was developed specifically for this study and has not been published before. The linking method was based on previously published tracking concepts for multi-target surveillance applications.^{8,9}

METHOD 2: DETECTION

A straightforward non-adaptive detection scheme includes a convolution step with a notional disk (in 2D) or spheroid (in 3D) shaped object model, followed by a local-maximum detector applied at the pixel

level. For this study, the radius of the shape model was typically set to 2 pixels in x and in y , and 1 pixel in z . Local maxima are those pixels (2D) or voxels (3D) with intensity larger than or equal to all neighbors (8-connected in 2D and 26-connected in 3D) and larger than at least one neighbor. After sorting the local maxima based on intensity, from highest (most likely to be a true particle) to lowest (least likely to be a true particle), a threshold is applied to result in a number of detections that exceeds by 25% the expected particle count based on aggregate ground-truth analysis of the data from the training phase. This simple scheme avoids the need for measurement clustering. Some redundant detections on extended particles may occur, but tend to be rare in practice. Our analysis of the intensity of detected particles on a per-image basis suggested an adaptive scheme to improve detection performance. For each image of a sequence, starting with a number of (highest-intensity) detections that is 50% of the expected number of particles, detections (with successively lower intensities) are added one by one, while keeping track of the mean intensity of the selected detections and the mean intensity of the non-selected detections. The process stops when the intensity of the present candidate detection has equal distance to these two means. If this does not occur, the scheme reverts to the non-adaptive one, and the number of detections is set to 125% of the expected number of particles.

METHOD 2: LINKING

The linking of detected particle objects is done by means of multiple hypothesis tracking (MHT). This powerful tracking paradigm was first formalized in what is now referred to as the hypothesis-oriented approach.¹⁰ Unfortunately, this approach typically leads to an unmanageable number of hypotheses, even for small problems. In this study, we therefore employed the track-oriented MHT approach instead.^{9,11} The goal is to estimate the state history (tracks) X^k of all particles in the field of view, based on the sequence $Z^k = (Z_1, \dots, Z_k)$ of sets of detections Z_t , for all time points $t = 1, \dots, k$. Each object may exist for only a subset of these times, with a single birth and a single death occurrence, that is objects do not reappear. We introduce the auxiliary discrete state history q^k , which represents a full interpretation of all detection data: which detections are false, how the object-originated ones are to be associated, and when objects are born and die. Two fundamental assumptions are made here: 1) there are no object births in the absence of a corresponding detection (we do not reason over new, undetected objects), and 2) there is at most one detection per object per image of a sequence.

We are interested in the probability distribution $p(X^k|Z^k)$ of object state histories, given the detections. This quantity can be obtained by conditioning over all possible auxiliary state histories:

$$p(X^k|Z^k) = \sum_{q^k} p(X^k|Z^k, q^k) p(q^k|Z^k).$$

The MHT approach aims to compute the maximum a posteriori (MAP) estimate for the auxiliary state history q^k , and the corresponding minimum mean-squared error (MMSE) estimate for the object state history X^k , conditioned on the estimate for q^k :

$$\hat{q}^k = \arg \max_{q^k} p(q^k|Z^k),$$

$$\hat{X}^k = \hat{X}_{\text{MMSE}}^k(Z^k, \hat{q}^k).$$

Compared to hypothesis-oriented MHT, track-oriented MHT avoids enumeration of all global hypotheses q^k , though these are implicitly defined in the set of track hypotheses trees. In practice, computational requirements preclude optimal, batch MHT solutions. Practical MHT solutions adopt a number of computational simplifications that include the following: 1) Hypotheses over undetected object births are excluded. 2) Hypothesis growth is limited by hypothesis gating, which disallows sufficiently unlikely associations, using the standard chi-square test with elliptical gating.¹² Following Blackman's notation, the threshold is set to $G = 9.21$, which leads to 99% gating. That is, statistically, 1% of particle-originated measurements will not be considered feasible updates to an active track on the same particle. Further, we do not spawn both missed-detection and death hypotheses. Generally, only a missed-detection hypothesis is spawned, and after a sufficient number of missed detections only an object-death hypothesis is spawned. 3) The track hypothesis trees are pruned so as to result in a single global hypothesis with a fixed latency in terms of number of images. For example, if processing is done with a two-image tree depth, particle tracks at time k are reported only after processing of detections at time $k + 2$. Hypothesis pruning generally relies on relaxation techniques for an otherwise challenging integer optimization problem. Well-known techniques include Lagrangian relaxation and linear programming.^{8, 13, 14} 4) Track extraction is performed sequentially using logic-based or statistical tests including the sequential probabilistic ratio test (SPRT).¹⁵ We opted for the former approach as in practice the SPRT is generally quite similar to the M -of- N confirmation test and K -miss termination test.

The key enabler for track-oriented MHT is the recursive formulation for the posterior probability of a global hypothesis given below. This expression relies on the assumption that in each image, the number of object births is Poisson distributed with mean λ_b (the discrete-time equivalent to the continuous-time birth rate), the number of false returns is Poisson distributed with mean λ_a (though this number does

not impact track extraction), objects die with probability p_χ (the discrete-time equivalent to the continuous-time death rate), and objects are detected with probability p_d . Further, τ is the number of tentative tracks under q^{k-1} and, with q^k , d is the number of detections, χ is the number of track deaths, and b is the number of new tracks. Correspondingly, if r denotes the number of detections in the current image, $r - d - b$ is the number of false returns. Finally, J_d , J_b , and J_a , denote the sets of detections, births, and false alarms, respectively, with $|J_d| + |J_b| + |J_a| = r$. Thus:

$$p(q^k|Z^k) = p_\chi^\chi \left((1 - p_\chi)(1 - p_d) \right)^{\tau - \chi - d} \cdot \prod_{j \in J_d} \left(\frac{(1 - p_\chi)p_d f_d(z_j|Z^{k-1}, q^k)}{\lambda_a f_a(z_j|Z^{k-1}, q^k)} \right) \\ \cdot \prod_{j \in J_b} \left(\frac{p_d \lambda_b f_b(z_j|Z^{k-1}, q^k)}{\lambda_a f_a(z_j|Z^{k-1}, q^k)} \right) \cdot \frac{p(q^{k-1}|Z^{k-1})}{\bar{c}_k},$$

with

$$\bar{c}_k = c_k \cdot \left(\lambda_a^r \frac{\exp(-\lambda_b - \lambda_a)}{r!} \prod_{j \in J_d \cup J_b \cup J_a} f_a(z_j|Z^{k-1}, q^k) \right)^{-1}.$$

The expressions given here allow for quite general object and sensor models. In the context of this study, standard linear Gaussian assumptions were used, for which the Kalman filter is directly applicable. Thus, $f_d(z_j|Z^{k-1}, q^k)$, that is the probability of observing z_j given a sequence of preceding measurements, is a Gaussian residual. This quantity relies on the Kalman filter prediction step with an assumed kinematic process noise that accounts for object maneuvering uncertainty as well as the assumed measurement error standard deviation in each dimension (x , y , and z if applicable). If there is no prior information on the objects, $f_b(z_j|Z^{k-1}, q^k)$ is the value of the uniform density function over measurement space. Similarly, $f_a(z_j|Z^{k-1}, q^k)$ is taken to be the value of the uniform density function over measurement space, under the assumption of uniformly distributed false returns. Depending on the scenario, either the near-constant velocity or near-constant position model was used. Filter initialization in the case of the nearly-constant velocity model relies on a zero-mean prior velocity assumption with known velocity variance in each dimension (x , y , and z if applicable).

Tracking performance improves substantively if track extraction processing impacts data association processing. Thus, the recursive track scoring expression noted above is modified slightly by introducing two track-update score multipliers (confirmation reward and continuation reward) greater than unity that increase $f_d(z_j|Z^{k-1}, q^k)$ for nearly-confirmed and confirmed tracks. They are chosen to be greater

for confirmed tracks than for nearly-confirmed tracks. Correspondingly, confirmed tracks are favored over nearly-confirmed tracks, and nearly-confirmed tracks are favored over tentative tracks.

METHOD 2: PARAMETERS

The user parameters of this method (and typical values) are:

- **Radius** of the shape model for detection (2 pixels in x - y and 1 pixel in z).
- Parameters **M** and **N** used in the M -of- N confirmation test (3-10).
- Parameter **K** used in the K -miss termination test (1-3).
- **Confirmation reward** multiplier used in the track scoring (2).
- **Continuation reward** multiplier used in the track scoring (4).

Other parameters of this method automatically computed from training data:

- **Expected particle count** for detection thresholding.
- **Birth rate** of new particles in the field of view.
- **Death rate** of existing particles in the field of view.
- **Detection probability** of particles in the field of view.
- **Kinematic process noise** used in the Kalman filter.
- **Measurement error** standard deviation used in the Kalman filter.
- **Velocity variance** used for Kalman filter initialization.

The exact parameter values for each image sequence in this study are given in **Supplementary Table 1**.

METHOD 3

Authors: Nicolas Chenouard, Fabrice de Chaumont, Jean-Christophe Olivo-Marin

Email: jcolivo@pasteur.fr

Software: Contact the authors

Form: Icy plugin

METHOD 3: INTRODUCTION

The particle detection procedure of this method is based on a previously published method¹⁶ for extracting spots in biological images that exploits the multiscale nature of the particle signal compared to the noise. The linking procedure has also been published recently^{17,18} and was designed to deal with missing and spurious detections by multiframe optimization using multiple hypothesis tracking.

METHOD 3: DETECTION

The particle detection step consists in a signal enhancement based on image filtering with a wavelet transform, followed by shrinkage of the wavelet coefficients, image reconstruction, and particle identification in the particle-enhanced image. Multiscale image decomposition is performed using an undecimated wavelet transform based on B-splines.¹⁹ This transform is well-suited for the task of fluorescent particle enhancement, as it well-approximates Gaussian functions of various sizes that are known to be good approximations of the point spread function (PSF) of microscopes, and the band-pass properties of the filters cancel out constant and slowly varying backgrounds. Relevant coefficients for the particle detection task are identified by applying a hard threshold to the amplitude of the coefficients at each scale. The rationale for this step is that in the wavelet domain, particle signals are sparsely represented (few coefficients are sufficient to represent particles), while noise is spread out. Thresholding the wavelet coefficients thus largely discards noise and preserves the particle coefficients. After reconstruction of the image from the thresholded coefficients, noise should thus be largely removed and the background is flattened. This allows us to identify particle locations simply as local intensity maxima. In this study we used a two-scale wavelet decomposition, a coefficients threshold of 80 to 100 depending on SNR and particle density, and a minimal distance of 2 pixels between detections to avoid multiple detections per particle.

METHOD 3: LINKING

After putative particles have been detected in the whole image sequence, they need to be linked through time into tracks, missing detections need to be compensated for, and spurious detections need to be discarded. To this end, an optimization framework was used that aims at solving these issues all at the same time, via multiple hypothesis tracking (MHT).^{17,18} It relies on an exhaustive Bayesian model of the tracking problem that is used to estimate the likelihood of the tracks. The model includes a statistical motion model that can account for both diffusive and directed particle movements, or for one of these alone. The statistics of the detection stage (the number of false and missed detections), as well as the statistics of particle appearance, are also included in the model. In this study, these statistics were learned from the training data, which was representative for the data used in the competition phase. In the case of real biological data, the statistics will need to be estimated manually from a representative subset of the data. The tracking problem is recast as a frame-to-frame likelihood maximization problem, with the specific feature of including both past and several future time points for each optimization step. Multiframe tracking helps resolving the correspondence between tracks and detections, discarding false

detections and compensating for missing ones, especially in low SNR conditions.¹⁸ Each likelihood optimization step is formulated as a combinatorial tree exploration that can be efficiently solved using dedicated hypothesis pruning strategies.¹⁷ No further processing is required after solving the likelihood maximization problem for each frame, since track creation and the identification of spurious and missing detections are all natively taken into account in our framework.

METHOD 3: PARAMETERS

The user parameters of this method (and typical values) are:

- **Wavelet coefficient threshold** (80-110).
- **Particle detection probability** (0.70-0.99).
- **Expected number of false detections** per frame (0.1-100 depending on SNR).
- **Track confirmation threshold** based on particle probabilities (0.1-0.5).
- **Track termination threshold** based on particle probabilities (0.001-0.01).
- **Expected track length** (fixed to 25 frames in this study).
- **Expected motion type** (diffusive, directed, or mixed, as appropriate).
- **Motion updating** based on estimated or given diffusion coefficient (not used here).
- **Expected particle displacement** between frames (3-5 pixels in x - y and 1.5 slices in z).
- **Gate factor** determining the search area for particle association (3-5).
- **Number of future frames used** in the likelihood maximization (3-4).
- **Expected number of new particles** per frame (5-100).
- **Expected number of initial particles** in the first frame (100-400).

The exact parameter values for each image sequence in this study are given in **Supplementary Table 1**.

METHOD 4

Authors: Mark Winter and Andrew R. Cohen

Email: acohen@coe.drexel.edu

Software: Contact the authors

Form: Matlab script

METHOD 4: INTRODUCTION

This method is based on multitemporal association tracking (MAT), which is a solution to the problem of associating future paths of detection (or segmentation) results to extend the set of existing tracks. MAT

is an alternative to the bipartite or multidimensional assignment approaches typically used with multiple hypothesis tracking (MHT). The main difference between MAT and approaches based on MHT is in the formulation of the optimization constraints in the data association, which should result in improved robustness to detection errors caused by noise or visual ambiguity in the images. MAT has been applied previously to tracking axonal organelle transport²⁰ and to tracking and lineaging proliferating cells.²¹ In MAT, an application-specific detection algorithm is applied to all image frames prior to tracking. There is also a post-tracking refinement step that uses the tracking results to improve the detection.

METHOD 4: DETECTION

Background noise removal is applied prior to detection. To this end, the observed image \tilde{I} is modeled as a combination of low-frequency background noise B , random shot noise R , and the original image I to be recovered: $\tilde{I} = I + B + R$.²² The low-frequency background contribution is estimated using a low-pass filter, implemented by iteratively convolving the image with a Gaussian kernel (with $\sigma = 1$ pixel). After subtracting the estimated background component from the observed image, the high frequency shot noise is removed using a median filter to produce the final denoised image. Detection of particles in the denoised image is accomplished by separately applying Otsu thresholding²³ and regional local maxima detection²⁴ and combining the results of these operations by a logical AND. Particle positions are taken as the centroid of regional maxima pixels in each connected component of foreground pixels in the denoised image. The detection algorithm has only two parameters: the number of iterations of convolution with the Gaussian kernel for the low-pass filtering, and the weighting factor for the Otsu thresholding on the denoised data. These two parameters were set automatically using the training data to guide the search of the range of parameter values. After the linking stage, described below, the detection results are refined by two additional steps. First, for each track, the location of the corresponding particle in each frame is adjusted by selecting the location on the regional maximum of the denoised image that minimizes the linear prediction error from the surrounding frames. Second, to accommodate occlusions, detections are added to each track for any frames where no detections were assigned, using linear interpolation on the locations of surrounding detections in time. Together, these refinements generated a few percent improvement in accuracy on the training data.

METHOD 4: LINKING

Solutions to the data association problem generally define an association function $\alpha(\tau_i, \rho_j)$, where τ_i is a track to be extended, and ρ_j is a sequence of detections extending W frames into the future, with

$\alpha(\tau_i, \rho_j) = 0$ if ρ_j is not assigned to τ_i , and $\alpha(\tau_i, \rho_j) = 1$ if it is. In MHT, the data association problem can be solved between adjacent frames (using, for example, bipartite or rectangular assignment), or across a window of frames simultaneously (multidimensional assignment). In either case, MHT implementations typically include the constraint that each track can be assigned to at most one sequence of detections, and vice versa: $\sum_i \alpha(\tau_i, \rho_j) \leq 1$ and $\sum_j \alpha(\tau_i, \rho_j) \leq 1$. To be able to deal with more challenging applications, where particles may (actually or apparently) split or merge, this constraint is relaxed in MAT. That is, a single track is allowed to be associated with more than one sequence of detections, and vice versa. In addition to capturing the visual aspects of the underlying biological problem where objects do appear to split and merge, this also changes the computational nature of the problem, allowing a solution to be optimized using minimum spanning tree algorithms in polynomial time. Importantly, MAT does not require explicit modeling of any characteristics except for typical behaviors of the objects being tracked. For Scenario 2 (microtubules) in this study, the typical behavior used by the tracker was the approximately constant velocity of particle motion.

MAT proceeds incrementally through each image frame. The first step is path discovery, which identifies all track extensions, or possible sequences of detections, extending up to W frames into the future. To improve performance, path discovery is gated using empirically determined maximum velocity and acceleration values. A track suitability, or cost function, assigns a cost to each possible track extension. By minimizing this cost function, MAT approximates the Bayesian a posteriori probability estimate for the data association problem.²⁵ For Scenario 2 in this study, the cost function was formulated as a weighted combination of linear prediction along the track and extension. The prediction was done by computing the minimum norm solution to the linear least-squares problem using singular value decomposition. This cost function is a simplified version of the cost function applied previously for tracking axonal organelle transport.²⁰ The function values for each track extension are preserved in a sparse graph and can be used in subsequent processing to incorporate higher-level knowledge of the application domain. In this study, these preserved costs were used in the post-tracking refinement of the detections, described above. After path discovery, tracks are associated with extensions using a minimum spanning tree algorithm.²⁶ Occlusions are handled by allowing tracks not associated with any extensions to be included in the processing for subsequent frames. The MAT algorithm has only two parameters: maximum velocity and acceleration. These were estimated empirically using the training data set and fixed for all image sequences in the competition phase of the study.

METHOD 4: PARAMETERS

The user parameters of this method (and typical values) are:

- **Number of iterations** of convolution with the Gaussian kernel (10-500).
- **Weighting factor** for the Otsu thresholding (1-3).
- **Temporal window size** for multidimensional assignment (4 frames).
- **Maximum velocity** used to constrain the linking (10 pixels/frame).
- **Maximum acceleration** used to constrain the linking (7.5 pixels/frame²).

The exact parameter values for each image sequence in this study are given in **Supplementary Table 1**.

METHOD 5

Authors: William J. Godinez and Karl Rohr

Email: k.rohr@dkfz-heidelberg.de

Software: Contact the authors

Form: Java module

METHOD 5: INTRODUCTION

This method is based on our previously published work^{27,28} and uses a probabilistic approach to particle tracking that relies on particle representation via Gaussian functions, detection and localization of particles based on the spot-enhancing filter or Gaussian fitting, position estimation of individual particles using Kalman filters, and measurement integration via probabilistic data association. More concretely, each particle is represented by either a 2D isotropic Gaussian function, a 2D anisotropic Gaussian function, or a 3D anisotropic Gaussian function, depending on the application. The Gaussian function is parameterized by the position $\mathbf{p} = (x, y)$ or $\mathbf{p} = (x, y, z)$ of the particle in a 2D or 3D image, respectively, by the peak intensity I_{\max} , and by the standard deviations σ_x, σ_y (2D case) or $\sigma_x, \sigma_y, \sigma_z$ (3D case). For each image of an image sequence, the method performs four steps: 1) detection and localization of particles, 2) prediction of the position of the tracked particles, 3) matching of the predicted positions with the detected particles, and 4) position estimation.

METHOD 5: DETECTION

To detect and localize particles, image regions corresponding to the objects of interest are enhanced using the spot-enhancing filter, also known as the Laplacian-of-Gaussian filter.^{29,30} The standard deviations σ_{Fxy} and σ_{Fz} of the filter are chosen based on the size of the particles (typical values are 2

pixels and 1.5 voxels, respectively). To detect regions corresponding to particles, a threshold for the filter response is computed and a connected-components labeling algorithm (8-connectivity in 2D and 6-connectivity in 3D) is applied. The threshold for the filtered image intensities is automatically determined based on the mean intensity of the filtered image plus a factor $|c|$ (typically 3) times the standard deviation of the filtered image intensities. The position of each particle is determined by computing the intensity-weighted center of mass. For 2D image data of elongated objects (such as microtubule tips), the underlying 2D Gaussian representation takes into consideration the orientation of the object. To estimate the orientation of the object we use a 2D Gaussian fitting scheme.²⁷ To reduce the image noise, a Gaussian filter is used. The value for the standard deviation σ_{Fxy} of the filter is set based on the size of the particles. To suppress the background, intensity values below the clipping threshold T_{clip} are set to T_{clip} . The clipping threshold is computed as the mean intensity of the image plus a factor $|c|$ times the standard deviation of the image intensities. Image regions corresponding to particles are detected by performing a search for local intensity maxima on the clipped image. To localize each particle, we fit a 2D Gaussian function to each candidate image region.

METHOD 5: LINKING

To calculate a prediction for the position of each particle we use a spatial-temporal filter. In cases where particles perform a random walk, we use a Kalman filter with a random walk motion model. In cases where particles move in a directed manner, we use a Kalman filter with a constant-velocity motion model. In cases where particles alternate between random walk motion and directed motion, we use an interacting multiple model (IMM) filter^{31,32} using a random walk model as well as a constant velocity model. The random walk motion model is parameterized by the expected squared displacement q_{RW} (typically 9 pixels²/frame²) of the particle over a time interval while the constant-velocity model is driven by the expected squared deviation of each velocity component q_{CV} (typically 25 pixels²/frame³) over a time interval. For finding the correspondence between a predicted position and a detected particle we use a global nearest-neighbor approach.⁷ In the case of the IMM filter, the correspondence approach considers only the predicted position of the motion model with the highest a priori probability. To improve the robustness, the spatial-temporal filter queries the image directly at positions determined by the predicted image position and its associated covariance matrix, as computed by the Kalman filter. Each position is assigned a weight that reflects the likelihood that the tracked particle generated a 2D or 3D Gaussian-like blob at the evaluated image position.²⁸ The likelihood is regulated by the expected level of image noise σ_n (typically 12 intensity units). This weight is also calculated for the predicted and

detected positions. To take into account the presence of neighboring objects, for each tracked object we compute the support provided by each image position to its neighboring objects as reflected by the weights of the neighboring objects. Using this information, the weights are adjusted by penalizing the weights of positions where the support for the neighboring objects is high. The multiple positions and their corresponding weights are integrated into the Kalman filter via a combined innovation.²⁸ For the IMM filter, the weights are also used to calculate the likelihood of each motion model. The final position of a tracked particle is computed using the associated spatial-temporal filter.

METHOD 5: PARAMETERS

The user parameters of this method (and typical values) are:

- **Standard deviation** (σ_{Fxy} and σ_{Fz}) of the Laplacian-of-Gaussian filter (2 pixels and 1.5 voxels).
- **Threshold factor** (c) for particle detection (-3) and background suppression (-0.5).
- **Expected square displacement** (q_{RW}) in the random walk motion model (9 pixels²/frame²).
- **Expected square deviation** (q_{CV}) in the constant-velocity motion model (25 pixels²/frame³).
- **Expected noise level** (σ_n) in the likelihood calculations (5-50 intensity units).

The exact parameter values for each image sequence in this study are given in **Supplementary Table 1**.

METHOD 6

Author: Yannis Kalaidzidis

Email: kalaidzi@mpi-cbg.de

Software: <http://motiontracking.mpi-cbg.de/>

Form: Windows executable

METHOD 6: INTRODUCTION

This method was developed specifically for tracking particles in 2D fluorescence microscopy image sequences and combines an efficient particle fitting algorithm with dynamic programming for linking. A detailed description of the method and its application to the study of endosome biogenesis and maintenance mechanisms has already been published.³³

METHOD 6: DETECTION

Prior to particle detection a background subtraction step is performed. The background is estimated using a windowed floating mean algorithm. Briefly, the median value within a sliding window is

calculated, and all values above the median plus two times the standard deviation are excluded from the set, after which the median value is recalculated. The process is repeated three times. This approach avoids overestimating the background as a result of extremely bright objects. The estimated background is subtracted from the image, and the remaining image signal with fluorescent particles is modeled by the following expression:

$$I(x, y) = \sum_i \frac{A_i}{1 + \left\{ \left[\frac{(x - x_i) \cos(\alpha_i) - (y - y_i) \sin(\alpha_i)}{w_i} \right]^2 + \left[\frac{(x - x_i) \sin(\alpha_i) + (y - y_i) \cos(\alpha_i)}{h_i} \right]^2 \right\}^2} + B_i$$

where A_i is the particle intensity, (x_i, y_i) is the particle center position, w_i and h_i are, respectively, the width and height of the particle, α_i is the angle between the principal axes of the particle and the image axes, B_i is the background residue in the vicinity of the particle, and i is the particle index. The use of the Lorentzian function $1/(1 + z^2)$ for modeling the particle profiles was inspired by practical arguments. It is qualitatively similar to using a Gaussian but computationally much faster. Fitting of the model is done using the modified Powell algorithm.³⁴ Fitted structures are subtracted from the image, and the procedure is repeated iteratively to capture all structures above two standard deviation of the noise. The latter is assumed to be Poisson distributed and its parameters are estimated from the background in the image sequence. Two partially overlapping particles are combined if the ratio between the local minimum intensity and the intensity of the smallest of the two peaks exceeds 40-90% (a user-definable parameter). The fitting produces a set of particles with known position, cross-sectional area, and intensity statistics.

METHOD 6: LINKING

The linking step is performed using dynamic programming, assigning four consecutive image frames at once, with track breaks as assignment possibility, and overcoming the use of greedy algorithms. Track assignments maximize the weighted sum of costs for position, speed, cross-sectional area, peak intensity, sum intensity, and deviation from a straight line, using a previously published approach.³⁵ The user-definable weights (not normalized) in this procedure were manually optimized per image sequence.

METHOD 6: PARAMETERS

The user parameters of this method (and typical values) are:

- **Window size** for estimating the background (60 x 60 pixels).
- **Merging ratio** of intensities for overlapping particles (40-90%).
- **Position weight** in the cost function for track assignment (0.7).
- **Speed weight** in the cost function for track assignment (0.01-0.3).
- **Area weight** in the cost function for track assignment (0.3-0.6).
- **Peak intensity weight** in the cost function for track assignment (0.15-0.3).
- **Sum intensity weight** in the cost function for track assignment (0.01-0.1).
- **Deviation weight** in the cost function for track assignment (0.01-0.5).

The exact parameter values for each image sequence in this study are given in **Supplementary Table 1**.

METHOD 7

Authors: Liang Liang, James Duncan, Hongying Shen, Yingke Xu

Email: liang.liang@yale.edu

Software: Contact the authors

Form: Matlab script

METHOD 7: INTRODUCTION

This method is based on a previously published method,³⁶ originally developed for the analysis of clathrin mediated endocytosis by tracking particles, namely clathrin coated pits (CCPs). The most important bio-parameters are the lifetimes of the particles, and the absolute positions are less interesting. Quantification of the lifetime parameters requires high detection and linking accuracies, as expressed by the numbers of the true/false positives/negatives, or related measures such as the Jaccard similarity index. For the present study, the method was further equipped with an interacting multiple model (IMM) filter and integer programming, and applied to Scenarios 1 (vesicles) and 3 (receptors).

METHOD 7: DETECTION

Particles are assumed to have a Gaussian profile on top of a constant background intensity level and possessing Poisson noise statistics. Potential particle loci are detected frame-by-frame by using the normalized Laplacian-of-Gaussian (LoG) filter^{29,30} and finding the local maxima in the filter response. To separate true-positive from false-positive (noise) detections, an optimal threshold is automatically

determined by fitting a two-component Gaussian mixture model to the distribution of the original image intensities and of the potential particle loci, using an expectation maximization algorithm. A subpixel position estimate of each surviving detection after thresholding is obtained by fitting Gaussian mixture models to the original image data at these loci. Background intensity estimation is used in this process as it improves the accuracy of the fitting.³⁷ The parameters (and typical values) of the detection algorithm are the standard deviation of the LoG filter reflecting average particle size (about 1-3 pixels), the particle SNR threshold (about 3-5) used to adjust the initial detection results, and the range of particle radii (1-2 pixels) used to constrain the Gaussian mixture model fitting. The optimal parameter values for each image sequence were estimated by fitting Gaussian functions to isolated particles. Here, the SNR of a particle is defined as the mean intensity of the particle divided by the standard deviation of the image noise, and therefore we can select the detected particles by SNRs before fitting the Gaussian functions to get their peak intensities. It is different from the definition of SNR in this study (where the peak intensity is used to calculate SNR) and is usually much smaller than the nominal SNR of each dataset.

METHOD 7: LINKING

A multiple hypothesis tracking (MHT) based approach is used in the linking stage to find the optimal tracks of detected particles.³⁶ It can be formulated and solved as an integer programming problem.³⁸ In cases where particles show Brownian motion (Scenario 1), each particle is represented as a state vector (x_t, y_t, I_t) , where (x_t, y_t) denotes the position at time (frame) t , and I_t is the intensity. Correspondingly, the cost function used for linking detected particles to tracks contains two terms: one related to particle displacement and the other related to intensity difference, as detailed elsewhere.³⁶ If the intensities of the particles do not change that much, the weight for the second term can be set to a very low value (0.01), effectively causing intensity information to be ignored. In cases where particles switch between Brownian and (nearly) constant velocity motion (Scenario 3), each particle is represented as a state vector $(x_t, v_t^x, y_t, v_t^y, I_t)$, where (v_t^x, v_t^y) is the velocity at time t . To accommodate the abrupt switching between the two motion models, an interacting multiple model (IMM) filter is used.^{39,40} A newly developed three-round tracking strategy is used to resolve the problem of (visual) particle splitting and merging. In the first round, tracking is performed in the forward time direction, and a merge-detection module detects if two or more particles are cluttered. If so, redetection is performed to the cluttered regions by considering temporal information. In the second round, tracking is performed in the backward time direction to resolve the splitting problem. And in the third round, tracking is performed in the forward time direction again to refine the result. The process noise covariance matrix for the two motion

models, the model transition probability matrix, as well as the radius of the spatial search window used to reduce the number of link candidates, were estimated from tracks obtained by first running a simple nearest-neighbor linking algorithm on well-separated particles in the images.

METHOD 7: PARAMETERS

The user parameters of this method (and typical values) are:

- **Standard deviation** (sigma) of the LoG filter (1-3 pixels).
- **SNR threshold** of the particles to be detected (3-5).
- **Radius range** of the particles to be detected (1-2 pixels).
- **Search radius** for linking detected particles (5-10 pixels).

The exact parameter values for each image sequence in this study are given in **Supplementary Table 1**.

Note: The large differences in the receptors case (Scenario 3) between the results from the competition phase of the study and the results from the verification phase (see **Supplementary Table 3**) are caused by an error in the conversion of the Matlab output data files to XML in the submission stage, for which the authors took responsibility. The verified results were not used in the analysis.

METHOD 8

Authors: Klas E. G. Magnusson, Joakim Jaldén, Helen M. Blau

Email: klasma@kth.se

Software: Contact the authors

Form: Matlab script

METHOD 8: INTRODUCTION

This method was originally developed to track cells in transmission light microscopy such as bright-field, phase-contrast and differential interference contrast (DIC) imaging. The problem of tracking cells is different from tracking particles in that the motion of cells may be more complex and unpredictable and that cells are more likely to influence each other's tracks. Because of this and the fact that cells, unlike the particles in this study, can be distinguished based on their appearance, the algorithm focuses on data association and uses a very simple motion model. The particle detection step of this method is based on a recently published preconditioning technique.⁴¹ For the linking of detected particles into tracks we use a modified version of our previous algorithm.⁴²

METHOD 8: DETECTION

For particle detection it is first necessary to estimate the PSF underlying the creation of the images in each scenario. It is assumed that in the absence of noise, an image I is obtained by convolution of a particle density image G with the PSF P , that is $I = G * P$. Taking the Fourier transform, denoted by $\mathcal{F}\{\cdot\}$, yields $\mathcal{F}\{I\} = \mathcal{F}\{G\}\mathcal{F}\{P\}$, so that an estimate of the PSF can be found as:

$$\hat{P} = \mathcal{F}^{-1} \left\{ \frac{\mathcal{F}\{I\}}{\mathcal{F}\{G\}} \right\}.$$

This estimator is optimal if the noise is additive white Gaussian. In this study, the noise was Poisson, so that the estimator was theoretically not optimal for this purpose, but it worked well nevertheless. In Scenario 2 (microtubules), the fact that the particles have different orientations was not taken into account, and therefore the estimated PSF was an average taken over all orientations. For the creation of G in estimating P for a given scenario, a sample image I was taken from the training data together with its ground-truth particle positions, and the intensities at the nearest discrete image coordinates around these positions in G were set according to the distance to the particle. For example, in 2D, for a particle at subpixel position $(k + \xi, n + \eta)$, with $k, n \in \mathbb{N}$ and $\xi, \eta \in [0, 1] \subset \mathbb{R}$, we set $G(k, n) = (1 - \xi)(1 - \eta)$, $G(k + 1, n) = \xi(1 - \eta)$, $G(k, n + 1) = (1 - \xi)\eta$, and $G(k + 1, n + 1) = \xi\eta$. \mathcal{F} and \mathcal{F}^{-1} were computed using the fast Fourier transform (FFT) with zero padding. And the estimated PSF was truncated to a 41 x 41 pixel matrix (2D cases) or a 21 x 21 x 9 voxel tensor (3D cases).

Assuming the above mentioned PSF-convolution model for the imaging process, the detection algorithm uses the following optimization to recover the particle distribution:

$$\text{minimize } \|\mathbf{y} - \mathbf{Ax}\|_2^2 + \beta \|\mathbf{x}\|_1 \text{ subject to } \mathbf{x} \geq 0,$$

where $\mathbf{y} = \text{vec}(I)$, $\mathbf{x} = \text{vec}(G)$, $\mathbf{Ax} = \text{vec}(G * P)$, and β is a free parameter, tuned as described below. This is a preconditioning problem with an L_1 penalty to favor a sparse solution.⁴¹ The optimization is solved using the multiplicative update rule:

$$\mathbf{x}_p^{(i+1)} = \frac{\max\{[\mathbf{A}^T \mathbf{y}]_p - \beta/2, 0\}}{[\mathbf{A}^T \mathbf{A} \mathbf{x}^{(i)}]_p} \mathbf{x}_p^{(i)} \text{ with } \mathbf{x}_p^{(0)} = 1,$$

for all pixels or voxels p . This scheme has been proven to converge to the global minimum.⁴¹ For the 2D cases, the number of iterations i was limited to 10, after which the result hardly improved for the considered images. In the 3D cases, only 5 iterations were performed, to reduce the run time of the algorithm. The output of the preconditioning algorithm is a recovered particle distribution image, which can subsequently be thresholded to produce a number of detections, or particle candidates, to be tracked. We use a global threshold τ and consider local maxima above the threshold to be individual detections. Even though the true particles are supposed to be point objects, we consider the connected component of the thresholded image around each local maximum to be an outline of that detection. Whenever there are multiple local maxima in the same connected component, a seeded watershed algorithm is applied, with seeds at the local maxima, to break the pixel regions into individual detections. In order to obtain subpixel positioning of the particles in the 2D scenarios, parabolas are fit in each dimension to the pixel values closest to the maxima in the preconditioned image, and the coordinates of the maxima of the parabolas are used as the particle coordinates. In the case of the 3D scenario, however, no subpixel particle positioning is done. The free parameter β and the global threshold τ were optimized by performing a grid search while minimizing the objective function $\text{FN} + 0.5 \cdot \text{FP}$, where FN is the number of missed particles and FP the number of false alarms. We chose to put more emphasis on FN than on FP, because our tracking algorithm is more sensitive to missed particles than false alarms. In the grid search, we only had to generate the grid explicitly for β , as the objective function can be easily evaluated for all thresholds, given the values at the local maxima of the preconditioned image. Generally, for higher SNRs, the optimal β will be lower and τ will be higher.

METHOD 8: LINKING

Linking starts with an empty set of tracks and adds tracks iteratively in a greedy fashion that gives the largest possible increase to a scoring function based on the particle appearance and motion in the added tracks. To find the optimal tracks to add, the algorithm uses a state-space diagram that represents all of the different states that a particle can be in, in the consecutive images. In the diagram, there is a state associated with every detection, representing the event that the track passes through that particular detection. In addition, there is also an idle state for every image, corresponding to the track not being present in that particular image. The states related to image I_t are connected to those of image I_{t+1} by

arcs with scores representing the log probability of a particle transitioning between the connected states. Arcs that connect two detection states correspond to particle movements. Arcs that go from an idle state to a detection state correspond either to the creation of a new track or to the connection of a new detection to the end of a preexisting track. Finally, arcs that go from a detection state to an idle state correspond either to the termination of a track or to the connection of a preexisting track to the end of the new track currently under creation. The idle states in consecutive images are connected by arcs that have zero cost, so that multiple track fragments can be added in the same iteration of the algorithm. The track fragments will always be in disjoint sets of images, however, as paths through the state-space diagram can pass through only one of the states associated with a particular image. Previously,⁴² all of the detection nodes in the diagram related to I_t were linked to all of the detection nodes related to I_{t+1} , but in the implementation used in the present study, only the three highest scoring incoming arcs and the three highest scoring outgoing arcs of each node are included. This decreases the computational complexity of the algorithm significantly, without affecting the performance negatively.

Once the state-space diagram has been created, the problem of finding the highest scoring track to be added to the set of preexisting tracks is solved by finding the highest scoring path through the state-space diagram. This is equivalent to a shortest-path problem, and given that the state-space diagram is a trellis graph, it can be solved using the Viterbi algorithm.⁴² A potential problem is that incorrectly constructed tracks can hinder the creation of correct tracks later on. To avoid this, we allow swapping, where a preexisting track is broken so that the beginning of the second half is linked to the end of the new track currently under creation. When this happens, the Viterbi algorithm switches from extending the new track, to appending new states to the end of the first half of the broken track.⁴²

In this study, the motion model used to compute the scores associated with particle movement was Brownian motion, where the probability density function of the particle position at time $t + 1$ was taken to be a Gaussian centered around the particle position at time t . In the 2D scenarios, the covariance of the Gaussian distribution was taken to be a scaled identity matrix, while in the 3D scenario (viruses), a non-diagonal covariance matrix was used instead, as there was a correlation between the Cartesian components of the particle displacements. The covariance matrices were estimated separately from the training data of every image sequence. To compute the scores associated with different particle counts in the detections, we trained classifiers to determine how likely detections were to contain different

numbers of particles. Here we used a classifier based on Gaussian discriminant analysis, instead of multinomial logistic regression as done previously,⁴² as this turned out to be both more accurate and faster. In the 2D scenarios, the features used for classification were the number of pixels in each detection, the sum of the preconditioned image taken over these pixels, the axis ratio of the pixel regions, the standard deviation of the x and y coordinates of the pixels, and the standard deviation of the pixel coordinates weighted by the intensity of the preconditioned image. In the 3D scenario, we used only the number of voxels and the sum of the preconditioned image in the voxels. A separate classifier was trained on the training data of every image sequence. The probabilities of particles appearing and disappearing, required for the scores on arcs connecting detection states to idle states, were estimated from the training data of each image sequence separately in this study.

METHOD 8: PARAMETERS

The user parameters of this method (and typical values) are:

- **Sparseness parameter β** in the particle detection step (0-30).
- **Number of iterations** in the detection optimization (10 in 2D and 5 in 3D).
- **Global threshold τ** for selecting relevant local maxima (0-40 intensity units).
- **Probability of particle appearance** used for linking (0.04-0.08).
- **Probability of particle disappearance** used for linking (0.04-0.08).

The exact parameter values for each image sequence in this study are given in **Supplementary Table 1**.

METHOD 9

Author: Perrine Paul-Gilloteaux

Email: perrine.paul-gilloteaux@curie.fr

Software: Contact the author

Form: ImageJ plugin

METHOD 9: INTRODUCTION

This method was originally designed to track objects (not necessarily spots) already identified in 2D frames and has been applied previously to particle tracking and analysis in high-speed atomic force microscopy image series.^{43,44} The detection part has been specifically designed for this study.

METHOD 9: DETECTION

Different algorithms are used for particle detection in 2D as compared to 3D. In the former case, an optional preprocessing step is first applied, consisting in Laplacian-of-Gaussian filtering as proposed previously for spot detection,²⁹ which yields a local minimum wherever the correlation with a Gaussian is maximum. The output is then inverted and scaled to the full dynamic range of an 8-bit image (values 0-255). In this study, preprocessing was applied only to image sequences with SNR < 4. The actual detection step consists in finding the local maxima, sorting them in descending order of intensity, applying a flood-fill algorithm to each of them using a fixed noise tolerance value (a free parameter with default value 10), and discarding lower maxima whose fill-regions touch those of higher maxima. This algorithm is readily available as the *Find Maxima* plugin of ImageJ (National Institutes of Health, Bethesda, MD, USA).⁴⁵ The output of the 2D detection step is a list of objects for each time point t , defined by their (x, y) positions (with pixel precision only), and their peak intensities I . In 3D, on the other hand, detection is accomplished by filtering the image stack at each time point with a 3D Gaussian (of size $5 \times 5 \times 3$ voxels, with $\sigma_x = \sigma_y = 1.5$ pixels, and $\sigma_z = 0.5$ times the distance between slices), followed by thresholding at twice the value of the Otsu²³ level. The centroid of each 3D connected component is then fit to a 3D Gaussian using the maximization of the expectation of a Gaussian mixture model (only one Gaussian was used in this study). The output of the 3D detection step is a list of objects for each time point t , defined by their (x, y, z) positions (with subpixel precision), and the integrated intensity I of their corresponding connected component after thresholding.

METHOD 9: LINKING

The linking step of the method is based on a modified version of a previously published method.⁴⁶ It minimizes the global energy of associating detected particles over a whole image sequence. The set of tracks is initialized using a nearest-neighbor search within a given maximum search radius (a free parameter whose value is not very critical but mainly affects the computation speed). For each image frame, each detected particle is associated with its nearest neighbor (in terms of Euclidean distance) in the next frame. If the nearest neighbor has already been assigned, the two competing associations are compared based on Euclidean distance, and the one corresponding to the smallest traveled distance will be taken and the other rejected. To complete the initialization, an additional step of linkage of broken tracks is performed, by checking that there is no free neighbor at the next frame after the first association. After initialization, there can be free particles left, which will be tested in the second part of the algorithm. The energy of the resulting particle associations is then computed as follows. Define a

detected particle i at time t as the vector $\mathbf{p}_t^i = (x, y, z, I)^T$, with $z = 0$ in the 2D case, and assume that the elements are independent and normally distributed, with variance $\sigma_{x,y,z}^2$ for (x, y, z) and σ_I^2 for I . The probability of associating two detections, \mathbf{p}_t^i and $\mathbf{p}_{t+\tau}^j$, where τ is an integer number of frames ranging from 1 to D (the maximum allowed disappearance time), is then given by:

$$p(\mathbf{p}_t^i, \mathbf{p}_{t+\tau}^j) = \frac{1}{(2\pi)^2 |\mathbf{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{p}_t^i - \mathbf{p}_{t+\tau}^j)^T \mathbf{\Sigma}^{-1}(\mathbf{p}_t^i - \mathbf{p}_{t+\tau}^j)\right),$$

where the covariance matrix is $\mathbf{\Sigma} = \text{diag}(\sigma_{x,y,z}^2, \sigma_{x,y,z}^2, \sigma_{x,y,z}^2, \sigma_I^2)$ and $|\mathbf{\Sigma}|$ is its determinant. Defining the energy of an association as the negative logarithm of its probability:

$$E(\mathbf{p}_t^i, \mathbf{p}_{t+\tau}^j) = -\ln p(\mathbf{p}_t^i, \mathbf{p}_{t+\tau}^j),$$

we compute the global energy of a set of tracks as the sum of the energies of the individual associations. In this study we added a term to the global energy to favor longer tracks:

$$E_G = \sum_n \sum_t E(\mathbf{p}_t^n, \mathbf{p}_{t+1}^n) + N \cdot M,$$

where the summations extend over all tracks n and all relevant time points t , N is the total number of tracks, and M the maximum distance allowed. Minimization of the global energy is done by simulated annealing, which proposes and evaluates random changes from a set of possible changes. Changes can consist in assigning a detection in a future frame (within the maximum disappearance range) to deal with particle disappearance (or misdetection), or making a new association that replaces an existing one, or linking two broken tracks. If the energy difference ΔE due to a change is negative, the change is accepted. If it is positive, it is accepted with probability $\exp(-\Delta E/\psi^{(i)})$, where $\psi^{(i)}$ is the temperature for iteration i . The temperature starts at $\psi^{(0)} = 100$ and decreases according to $\psi^{(i+1)} = (1/i) \psi^{(i)}$. This means that even local changes that increase the global energy can be accepted, in particular in the first iterations, when the temperature is still high. If the global energy does not decrease, the tracks are reset to their previous state, and the next iteration is started. If it does decrease, the changes are all accepted, and the iteration count is reset to 0. The algorithm ends after i_{\max} iterations (a user-definable parameter). Notice that if $i_{\max} = 0$, the second part of the linking algorithm will not be executed, and the result will simply be the output of the initial nearest-neighbor search. In this study, of all resulting tracks, only the ones having a length of at least 3 frames were kept.

METHOD 9: PARAMETERS

The user parameters of this method (and typical values) are:

- **Standard deviation** (sigma) of the Laplacian-of-Gaussian filter (1.5-2.5 pixels).
- **Tolerance level** used in the flood-fill algorithm (20-60 depending on SNR).
- **Search radius** used in the nearest-neighbor algorithm (10-15 pixels).
- **Disappearance time** allowed in the linking algorithm (1-3 frames).
- **Number of iterations** in the simulated annealing algorithm (mostly 100).

The exact parameter values for each image sequence in this study are given in **Supplementary Table 1**.

METHOD 10

Authors: Philippe Roudot, Charles Kervrann, François Waharte

Email: philippe.roudot@inria.fr

Software: Contact the authors

Form: C++ code

METHOD 10: INTRODUCTION

This method was designed for tracking rotationally symmetric particles frame by frame in 2D image sequences. It was applied only to the image sequences of Scenario 1 (vesicles) of the present study. The detection step yields only approximate particle positions, estimated using the structure tensor⁴⁷ and optimal histogram based thresholding,⁴⁸ which are subsequently refined and linked using a Gaussian fitting scheme based on iterative reweighted least-squares minimization.⁴⁹

METHOD 10: DETECTION

The detection stage deals only with the presence of particles and their approximate positions, since the Gaussian fitting scheme described in the linking section provides a refined localization and displacement estimate, and detects the disappearance of particles. Particles are detected using a two-step approach. First, pixels are classified as potential particle pixels using a gradient-based measure. The measure used in this study is based on the so-called structure tensor,⁴⁷ or second-moment matrix, whose elements are estimated by convolving the image with the first derivative of the Gaussian kernel ($\sigma = 1$ pixel). This can be done computationally very efficiently thanks to the separability of the Gaussian. The tensor is integrated within a small window (typically 7×7 pixels) and the resulting smallest eigenvalue is used as classification measure. If the current frame is not the first frame, pixels belonging to the support of a

particle successfully tracked by the linker, are not processed. Potential new particle positions are selected as the N pixels having the highest scores for the classification measure while respecting a distance of 8 pixels between these highest pixels. Here, N represents an overestimation of the expected number of particles, which is a parameter of the algorithm and was set to 120% of the corresponding values for the low, mid, and high density images in this study. In the second step of the algorithm, the initially detected particle pixels are further analyzed, in order to discriminate between correct and spurious detections. This is done by computing the threshold m that best separates the two populations in terms of their scores for the used classification measure. The optimal threshold is estimated from the histogram H of the scores by minimizing the Matusita distance:^{48,50}

$$\hat{m} = \arg \min_m \left(\sum_{i=m_{\min}}^m \left(\sqrt{H(i)} - \max_{k \in (m_{\min}, m)} \sqrt{H(k)} \right)^2 + \sum_{i=m}^{m_{\max}} \left(\sqrt{H(i)} - \max_{k \in (m, m_{\max})} \sqrt{H(k)} \right)^2 \right)$$

Pixels with values above the threshold are taken as the true particle positions, up to pixel precision. A refinement of the detected particle positions is performed in the linking stage.

METHOD 10: LINKING

The linking of detected particle positions between successive frames, as well as the refinement of these positions, is accomplished by a Gaussian fitting scheme. Given an initial particle position \mathbf{p}_t in frame t , the refined (subpixel) position and size is given by the following M-estimator:

$$(\hat{\mathbf{p}}_t, \hat{\gamma}) = \arg \min_{(\mathbf{p}_t, \gamma)} \sum_{\mathbf{p} \in W(\mathbf{p}_t)} \rho_{\eta} \left(I_t(\mathbf{p}) - (b + a G_{\gamma}(\mathbf{p} - \mathbf{p}_t)) \right)$$

where G_{γ} denotes the Gaussian with variance γ^2 representing the shape of the particle and centered at the origin, a denotes its amplitude, b is the background level which is considered locally constant, and $W(\mathbf{p}_t)$ is a 7×7 pixel window around \mathbf{p}_t . At frame t , the amplitude is taken as $a = I_{t-1}(\mathbf{p}_{t-1}) - b$ in the case of tracking, and as $a = I_t(\mathbf{p}_t) - b$ in case of position refinement. Minimization is accomplished by an iteratively reweighted least-squares (IRLS) procedure. The weight function ρ_{η} is taken to be the Leclerc function $\rho_{\eta}(r) = 1 - \exp(-r^2/\eta^2)$ with scale parameter $\eta = \lambda \sigma_{\text{res}}$ such that outliers that do not respect the assumed Gaussian error distribution are weighed down. Here, σ_{res} is defined as⁵⁰ $\sigma_{\text{res}} = 1.4826 \text{ median}_j(|r_j - \text{median}_k(r_k)|)$, where $\text{median}_n(\cdot)$ denotes the median value taken over all positions within the window $W(\hat{\mathbf{p}}_t)$ and $\{r_n\}$ is the set of residuals obtained by least mean square

estimation. In this study, the parameters were initialized as follows: γ was set to 2, b was set to the minimum local value in a 7×7 pixel window, and λ was fixed to 3.

METHOD 10: PARAMETERS

The user parameters of this method (and typical values) are:

- **Standard deviation** (sigma) for computing the Gaussian first derivatives (1 pixel).
- **Window size** for structure tensor integration and Gaussian fitting (7 x 7 pixels).
- **Initial standard deviation** used in the Gaussian fitting algorithm (2 pixels).
- **Expected number of particles** (120% of the specified numbers in this study).

The exact parameter values for each image sequence in this study are given in **Supplementary Table 1**.

METHOD 11

Authors: Ihor Smal and Erik Meijering

Email: i.smal@erasmusmc.nl

Software: Contact the authors

Form: Java module

METHOD 11: INTRODUCTION

This method was designed for general purpose tracking of fluorescent particles exhibiting random-walk or directed motion with possible switching between these modes of displacement. It was developed specifically for this study and has not been published before but is largely based on existing concepts. The detection step combines wavelet-based detection¹⁶ and Gaussian fitting.⁵¹ The linking of detected particles is done using an adapted version of an existing multiframe assignment framework.⁵²

METHOD 11: DETECTION

To efficiently localize particles of interest in the noisy images and deal with spurious detections, two detection algorithms were used, which were chosen depending on the scenario. Both algorithms make use of a preprocessing step to find candidate locations. In this step, each image in a sequence is first transformed using the multiscale isotropic undecimated wavelet transform,¹⁶ with three scales. Next, the wavelet coefficients of the second and third scales below a threshold (given below) are discarded, and the corresponding retained coefficients are summed to yield a reconstructed image with reduced noise and low-frequency background variations. In the first detection algorithm, applied to Scenarios 1 and 3,

the local maxima in the resulting images are taken as the candidate positions. To discriminate between true particles and noise, a 2D Gaussian fitting algorithm⁵¹ is applied to the original images at these maxima, which yields for each candidate position its subpixel coordinates (x, y) , amplitude I , and size, expressed as the standard deviation σ of the Gaussian fit. All positions for which $I \in [I_{\min}, I_{\max}]$ and $\sigma \in [\sigma_{\min}, \sigma_{\max}]$ are then accepted as true particle positions. Typical values used for the parameters (depending on the SNR) were $I_{\min} = 3-8$, $I_{\max} = 150$, $\sigma_{\min} = 0.3-0.45$, $\sigma_{\max} = 2.0-2.5$, and thresholding of the wavelet coefficients in the preprocessing step was done at a level of $3.0-3.2\sigma_W$, where σ_W is the standard deviation of the wavelet coefficients for a given scale. For the second detection algorithm, applied to Scenario 2, where the particles were larger and asymmetrical, typical threshold values were $2.0-3.5\sigma_W$. In the reconstructed image, the algorithm then finds the connected components (of pixels with intensities larger than 0), and classifies them as true particles (if they satisfy a minimum size requirement) or spurious detections (otherwise). The subpixel coordinates of the former are computed as the center of mass of the retained connected components in the reconstructed image.

METHOD 11: LINKING

For the linking of detected particles, an existing multiframe assignment framework⁵² was adapted by introducing application specific cost functions, which measure how similar two detections from different frames are. In a simple two-frame assignment case, the detections from one frame are "competing for" the detections from the subsequent frame, and the optimal solution is characterized by a minimal total cost. However, in the case of touching or crossing tracks, when two particles come very close to each other, the detection algorithm will often provide only one detection, to be assigned to both tracks at the same time, which violates one of the basic assumptions of common two-frame assignment algorithms (such as bipartite matching, the Auction algorithm, and the Hungarian algorithm) that each detection can be assigned to only one track. To deal with this, as well as temporary disappearance of particles (up to two frames), we considered the linking step as a four-frame assignment problem, to be solved sequentially when going through the frames of the entire image sequence. The proposed cost functions c_1 , c_2 , and c_3 used in the assignment algorithm⁵² reflect the three different modes of motion (random walk, near-constant velocity, and switching between these two modes) in the three considered scenarios in this study (Scenario 1, 2, and 3, respectively). Cost c_1 is the Euclidean distance between position \mathbf{p}_t^i of detection i in frame t and position $\mathbf{p}_{t+\tau}^j$ of detection j in frame $t + \tau$. Cost c_2 is the Euclidean distance between position $\mathbf{p}_{t+\tau}^j$ and the position predicted from linear extrapolation of the last displacement:

$\mathbf{p}_t^i + \tau(\mathbf{p}_t^i - \mathbf{p}_{t-1}^i)$. And cost c_3 is a weighted combination of c_1 and c_2 , given as $c_3 = (\mu_1 c_1 + \mu_2 c_2) / (\mu_1 + \mu_2)$, where $\mu_1 = \exp(-\|\mathbf{p}_t^i - \mathbf{p}_{t-1}^i\|)$ and $\mu_2 = \exp(-\|\mathbf{p}_t^i - \mathbf{p}_{t-1}^i - 4\|^2 / 2)$.

METHOD 11: PARAMETERS

The user parameters of this method (and typical values) are:

- **Threshold** of the wavelet coefficients (2-4 times the standard deviation of the coefficients).
- **Minimum intensity** of a fitted particle to accept it as a true detection (3-25 units).
- **Maximum intensity** of a fitted particle to accept it as a true detection (150-200 units).
- **Minimum size** of a fitted particle to accept it as a true detection (a standard deviation of 0.3-0.6 pixels in Scenarios 1 & 3 and an area of 5-15 pixels in Scenario 2).
- **Maximum size** of a fitted particle to accept it as a true detection (a standard deviation of 2.0-2.5 pixels in Scenarios 1 & 3 and an area of 200 pixels in Scenario 2).

The exact parameter values for each image sequence in this study are given in **Supplementary Table 1**.

METHOD 12

Author: Jean-Yves Tinevez and Spencer L. Shorte

Email: tinevez@pasteur.fr

Software: <http://fiji.sc/TrackMate>

Form: ImageJ/Fiji plugin

METHOD 12: INTRODUCTION

This method was inspired by the particle tracking method of Jaqaman and colleagues.⁵³ It considers the linking as a two-step linear assignment problem and has the power to close gaps by first forming low-cost track segments and then linking these segments. The method takes as input a set of detected particles with corresponding features to be used in the linking-cost minimization algorithm. In our implementation, particle detection is accomplished using an existing approach based on difference-of-Gaussian filtering and parabolic interpolation.⁵⁴

METHOD 12: DETECTION

For the detection of particles a blob detector based on the difference of Gaussians was used.⁵⁴ Given d the approximate expected particle diameter, determined by inspection of the image data, two Gaussian filters, with standard deviation $\sigma_1 = d / (1 + \sqrt{2})$ and $\sigma_2 = \sigma_1 \sqrt{2}$ respectively, are applied individually to

each image in a sequence. The result of the second filter (largest σ) is then subtracted from the result of the first filter (smallest σ). This yields a smoothed image with sharp local maxima, which are taken to be the candidate particle locations. The intensities at these locations are used as a quality feature of the candidate particles. If two candidates are found to be closer than the expected radius $d/2$, the one with the lowest intensity is discarded. To obtain subpixel localization of each detected particle, a parabolic interpolation scheme is used.⁵⁴ The quality feature is also interpolated using this scheme. Finding local maxima creates a large number of spurious detections. These are discarded by applying a threshold on the quality feature. The value of this threshold is set manually to match the SNR of the input image. Thresholded detections are then retained for subsequent particle linking.

METHOD 12: LINKING

The linking of detected particles was considered as a linear assignment problem (LAP) and solved using the method of Jaqaman and colleagues,⁵³ with modifications in the calculation of the linking costs. In this method, linking is performed in two steps. First, track segments are created from frame-to-frame particle linking, and then the track segments are linked to achieve gap closing. The mathematical formulation used for both steps is the same: a cost matrix is assembled containing all possible assignments and their corresponding costs, and the actual assignment is obtained by solving the matrix for minimal total cost. The two steps are described here in more detail.

In the first step, for each two consecutive frames at t and $t + 1$, each detection in the first frame is offered to link to any detection in the next frame, or not to link. This takes the form of an $(N + M) \times (N + M)$ matrix, where N is the number of detections in frame t , and M the number of detections in frame $t + 1$. The matrix can be divided into four quadrants: 1) the top-right quadrant ($N \times M$) contains the costs for linking a detection i in frame t to a detection j in frame $t + 1$; 2) the top-left quadrant ($N \times N$) contains the costs for a detection i in frame t to not create a link with next frame (yielding a segment stop); 3) the bottom-right quadrant ($M \times M$) contains the costs for a detection j in frame $t + 1$ to not have a link with the previous frame (yielding a segment start); 4) the bottom-left quadrant ($M \times N$) is an auxiliary block that is mathematically required by the LAP formalism,⁵³ and is built by taking the transpose of the top-right quadrant, replacing all non-blocking costs by the minimal cost. The solution to the resulting LAP can be found in polynomial time using the Munkres algorithm.⁵⁵

In the original paper,⁵³ the costs in the matrix depend on the distance between detection j in frame $t + 1$ and the position of detection i from frame t propagated into frame $t + 1$ using Kalman filtering, possibly weighted by the difference in intensity between the two detections. In the implementation used in the present study, Kalman filtering is not employed, but the user is offered to tune the costs by adding features. Specifically, the user is asked for a maximum allowed linking distance, and for a series of features, with corresponding weights. For two detections that may link, the distance D between them is calculated, and if it is larger than the maximum allowed distance, the link is forbidden and the cost is set to infinity (the blocking value). If not, then for each feature f in the map, with values f_1 and f_2 for the two detections, a penalty is calculated as $p = 3w|f_1 - f_2|/(f_1 + f_2)$, where w is the weight associated with the feature. This expression is such that $p = 0$ if the two feature values are the same, $p = 1$ if $w = 1$ and one of the feature values is twice the other, and $p = 2$ if one of the values is five times the other. All penalties are then summed to yield $P = 1 + \sum p$ and the cost is set to $C = (D \cdot P)^2$. The costs in the top-left and bottom-right quadrants of the matrix, associated with track segment termination and initiation, are all set to the blocking value, except along the diagonal, where the costs are set to $C^* = 1.05 \max(C)$, with C the costs in the top-right quadrant.⁵³ Without additional features, the C for any two detections is simply the square of D (since $P = 1$ in that case), and the LAP solution is the one that minimizes the sum of squared distances, which corresponds to the case of Brownian motion.⁵⁶ Adding features and weights allows to provide a more detailed definition of particle "resemblance", and which possible links are to be favored, which may improve tracking robustness. In this study, however, this was done only for Scenario 2, where the particles display an elongated morphology, and their orientations change only marginally from frame to frame. This was exploited by fitting an ellipse to the intensity distribution around a detection, yielding the preferred orientation of the particle, which was used as a linking feature.

The frame-to-frame linking step described above is repeated for all consecutive frame pairs in an image sequence. This yields a series of non-branching track segments. A track segment may start or stop because of a missing detection, or because of a merge or split event, which is not taken into account at this stage. In the second step, track segments are offered to link with each other, using the same LAP framework.⁵³ A new cost matrix is generated for this purpose, which reflects the following events: 1) linking of the end of a track segment to the start of any other track segment (a gap-closing event, where a link is created typically over two detections separated by a missed detection); 2) linking of the start of a track segment to any detection (except start or end) of any other track segment (a splitting event, where

a track branches into two subtracks); 3) linking the end of a track segment to any detection (except start or end) of any other track segment (a merging event, where two tracks merge into one); 4) no linking of any detection of any two track segments. As before, the way the costs are calculated was modified, reusing the feature penalties framework described above. In addition, the user must provide a maximal time difference over which linking will be considered. In this study, however, only gap-closing events were considered, and in some cases this was not even necessary.

METHOD 12: PARAMETERS

The user parameters of this method (and typical values) are:

- **Expected particle diameter** for detection (3-5 pixels).
- **Quality feature threshold** to reject spurious detections (0.5-4.5).
- **Maximum allowed distance** in linking detected particles (10-15 pixels).
- **Orientation feature weight** in linking elongated particles (10).
- **Maximal time difference** over which gap closing is considered (3-4 frames).

The exact parameter values for each image sequence in this study are given in **Supplementary Table 1**.

METHOD 13

Authors: Joost Willemse, Katherine Celler, Gilles P. van Wezel

Email: jwillemse@biology.leidenuniv.nl

Software: Contact the authors

Form: ImageJ plugin and Matlab script

METHOD 13: INTRODUCTION

This method was designed specifically for this study and has been applied to the quantitative analysis of dynamic protein localization in *Streptomyces*.⁵⁷ The method resembles the classical method of Crocker and Grier.⁵⁶ After basic image filtering the actual detection is performed by thresholding on intensity and particle size. Linking of detected particles is based on nearest-neighbor searching but with the possibility to deal with temporary disappearance of particles.

METHOD 13: DETECTION

To detect particles in the images, several filtering steps are applied to enhance the contrast between natural autofluorescence and the observed particle fluorescence. The initial step for noise reduction is a

Gaussian filter ($\sigma = 1$ pixel), which keeps the particles intact while the noise is diffused throughout the image. After this, two processing steps are possible, depending on the SNR in the image. For high SNRs, a top-hat filter is applied to detect and emphasize foci that are brighter than the background. The result is an image in which the background has been equalized and the foci of interest have been intensified. Based on this image, a thresholding procedure is applied, in which the cut-off intensity (above which objects are considered true particles) can be manually chosen by the user. Setting the threshold correctly is a vital part of the algorithm. In the competition, the chosen threshold values were 30 (for SNR = 7), 20 (for SNR = 4), and an automatically determined level (using the auto-thresholding function of ImageJ) with a constant added (for SNR = 1 or 2, where the constant was 10 for high-density, 12 for mid-density, and 15 for low-density images), respectively. Thresholded objects that are either too large (in this study 500 pixels or larger) or too small (less than the expected minimum size of typically 6-10 pixels depending on SNR) are filtered out. Objects located close to each other are separated using a watershed-based splitting operation, using the Euclidean distance transform. For low-SNR images, a similar approach is taken, leaving out the top-hat filter. In noisier images, this levels the background only slightly compared to the objects of interest, since these generally consist of several intensity peaks in nearby pixels. Finally, the subpixel position of each thresholded particle region is taken to be the center of mass.

METHOD 13: LINKING

To track detected particles from frame to frame, a modified nearest-neighbor algorithm is used. Initially, the image space is divided into blocks (of tunable size, in this study 1 pixel in 2D cases, and 1 voxel in 3D cases), using a regular grid. Starting with the first frame, each particle is labeled according to the block containing its coordinates. In the next frame, the same grid is created, and the particles in that frame too are labeled according to their corresponding block. Then, each particle is linked with the particle in the previous frame that is nearest in its neighborhood, defined by the surrounding blocks (initially ± 1 block in each dimension). If multiple particles are detected, the closest one is chosen, and if none is found, the neighborhood is iteratively extended in all dimensions (to $\pm n$ blocks, $n = 1 \dots N$, with N a user-defined maximum range, typically 10). If the particles cannot be tracked within this distance, a gap-closing step is applied. This ensures that if particles disappear in a certain frame, and then reappear again, tracking continues. Gap closing is performed by running the tracking procedure on frame $t - 2$, and then continuing backwards along frames until the particle is "found" again. Thus in a sense, particles are not tracked forwards, but backwards. The user may choose how many frames backwards to search. In this study the backwards tracking parameter was set to 3 frames. Disappearing of particles may occur due to

noise or if a particle moves out of focus. Newly appearing particles are labeled as new objects and tracked thereafter in subsequent frames. If particles are tracked forward to the same particle in the next frame, the event is identified as a fusion. This is done based on particle area. It was estimated that a fusion consisting of two particles has 1.5 times the average area of a single particle, and that a fusion of three particles has 1.5 times the area of a fusion of two particles. These larger objects can be tracked back to two or three parents if these are detected in the neighborhood. Splitting particles are also detected, with both daughter particles tracking back to the same parent based on directionality. This is based on the assumption that on short time scales, tracks follow a straight path. Finally, it is possible in this method to eliminate single detections or short tracks. In the present study, single detections and tracks shorter than 3 frames were filtered out.

METHOD 13: PARAMETERS

The user parameters of this method (and typical values) are:

- **Standard deviation** (sigma) of the Gaussian filter for noise reduction (1 pixel).
- **Intensity threshold** (20-30 units or auto-computed plus an offset of 10-15 units).
- **Minimum size** of the thresholded objects (6-10 pixels).
- **Maximum range** within which to find nearest neighbors (10 pixels).
- **Backward tracking range** within which to close gaps (3 frames).
- **Minimum track length** in numbers of frames (3 frames).

The exact parameter values for each image sequence in this study are given in **Supplementary Table 1**.

METHOD 14

Authors: Han-Wei Dan and Yuh-Show Tsai

Email: dnadann@gmail.com

Software: Contact the authors

Form: ImageJ plugin

METHOD 14: INTRODUCTION

This method has been published previously and used for tracking of secretory vesicles with various dynamic properties in living cells.^{58,59} More details of the method can be found in the cited papers. The mentioned parameters of the method were manually tuned per image sequence in this study.

METHOD 14: DETECTION

Prior to particle detection the images are first processed using a Gaussian filter (with $\sigma = 0.5$ pixel) in order to reduce noise without compromising the particle signal. Next, a Wiener filter is applied to deblur the images and to restore image resolution.⁶⁰ In the filtered images, pixels with a value lower than the minimum intensity of the foreground (a tunable threshold typically set to 10-20) are identified as background, and set to zero. To enhance particles in the resulting images, a top-hat transformation is applied,⁶¹ using a disk-shaped structuring element (with tunable radius typically set to 2-6 pixels). The particle regions are then detected by thresholding (with a fixed threshold value of 1) to obtain the segmented (binary) images. A morphological opening can be applied to cut connected particles. For each component in the resulting segmented images, the algorithm extracts various features to be used in the linking step, including area, maximum intensity, and coordinates.

METHOD 14: LINKING

Two methods are used to link detected particles from frame to frame. The first uses a simple nearest-neighbor approach. For each particle in frame t , the particles in frame $t + 1$ with a Euclidean distance of at most the maximum allowed tracking distance (a parameter of the algorithm typically set to 5-8 pixels), are considered as candidates for linking. To determine the best match, the similarity between the particle and each of the candidate particles is calculated, based on distance, area difference, maximum intensity difference, and difference in direction compared to the displacement of the particle from $t - 1$ to t .⁵⁹ If no suitable candidate can be found using the nearest-neighbor approach, a Kalman filter is applied, which predicts a possible position, and the described search for the best matching particle in the next frame is repeated starting from that position. The tracking method is essentially a 2D method. Tracking in 3D image sequences is done by first recording for each pixel the slice number having the maximum intensity, then using the average intensity projection to perform tracking in 2D and finding the centroid (x, y) particle positions, and finally taking for each particle the recorded slice number at the nearest pixel position as the z coordinate of particle.

METHOD 14: PARAMETERS

The user parameters of this method (and typical values) are:

- **Standard deviation** (sigma) of the Gaussian filter for noise reduction (0.5 pixel).
- **Intensity threshold** for detection after image preprocessing (10-20 units).
- **Structuring element radius** of the top-hat transform (2-6 pixels).

- **Maximum linking distance** in the nearest-neighbor tracker (5-8 pixels).

The exact parameter values for each image sequence in this study are given in **Supplementary Table 1**.

REFERENCES

1. Meijering E, Smal I, Danuser G. Tracking in molecular bioimaging. *IEEE Signal Processing Magazine* **23**, 46-53 (2006).
2. Saxton MJ. Single-particle tracking: Connecting the dots. *Nature Methods* **5**, 671-672 (2008).
3. Kalaidzidis Y. Multiple objects tracking in fluorescence microscopy. *Journal of Mathematical Biology* **58**, 57-80 (2009).
4. Meijering E, Dzyubachyk O, Smal I, van Cappellen WA. Tracking in cell and developmental biology. *Seminars in Cell and Developmental Biology* **20**, 894-902 (2009).
5. Rohr K, Godinez WJ, Harder N, Wörz S, Mattes J, Tvaruskó W, Eils R. Tracking and quantitative analysis of dynamic movements of cells and particles. *Cold Spring Harbor Protocols* **2010**, pdb.top80 (2010).
6. Meijering E, Dzyubachyk O, Smal I. Methods for cell and particle tracking. *Methods in Enzymology* **504**, 183-200 (2012).
7. Sbalzarini IF, Koumoutsakos P. Feature point tracking and trajectory analysis for video imaging in cell biology. *Journal of Structural Biology* **151**, 182-195 (2005).
8. Coraluppi S, Carthel C. Recursive track fusion for multi-sensor surveillance. *Information Fusion* **5**, 23-33 (2004).
9. Coraluppi S, Carthel C. Multi-stage multiple-hypothesis tracking. *Journal of Advances in Information Fusion* **6**, 57-67 (2011).
10. Reid D. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control* **24**, 843-854 (1979).
11. Kurien T. Issues in the design of practical multitarget tracking algorithms. In *Multitarget-Multisensor Tracking: Advanced Applications*, Bar-Shalom Y (ed.), Artech House, Norwood, MA, USA (1990).
12. Blackman SS. *Multiple-Target Tracking with Radar Applications*. Artech House, Norwood, MA, USA (1986).
13. Poore AP, Rijavec N. A Lagrangian relaxation algorithm for multidimensional assignment problems arising from multitarget tracking. *SIAM Journal on Optimization* **3**, 544-563 (1993).
14. Storms P, Spieksma F. An LP-based algorithm for the data association problem in multitarget tracking. *Computers and Operations Research* **30**, 1067-1085 (2003).

15. Blackman S, Popoli R. *Design and Analysis of Modern Tracking Systems*. Artech House, Norwood, MA, USA (1999).
16. Olivo-Marin J-C. Extraction of spots in biological images using multiscale products. *Pattern Recognition* **35**, 1989-1996 (2002).
17. Chenouard N, Bloch I, Olivo-Marin J-C. Multiple hypothesis tracking in cluttered condition. *Proceedings of the IEEE International Conference on Image Processing*, 3621-3624 (2009).
18. Chenouard N, Bloch I, Olivo-Marin J-C. Multiple hypothesis tracking in microscopy images. *Proceedings of the IEEE International Symposium on Biomedical Imaging*, 1346-1349 (2009).
19. Starck J-L, Fadili J, Murtagh F. The undecimated wavelet decomposition and its reconstruction. *IEEE Transactions on Image Processing* **16**, 297-309 (2007).
20. Winter MR, Fang C, Banker G, Roysam B, Cohen AR. Axonal transport analysis using Multitemporal Association Tracking. *International Journal of Computational Biology and Drug Design* **5**, 35-48 (2012).
21. Winter M, Wait E, Roysam B, Goderie SK, Ali RA, Kokovay E, Temple S, Cohen AR. Vertebrate neural stem cell segmentation, tracking and lineaging with validation and editing. *Nature Protocols* **6**, 1942-1952 (2011).
22. Michel R, Steinmeyer R, Falk M, Harms GS. A new detection algorithm for image analysis of single, fluorescence-labeled proteins in living cells. *Microscopy Research and Technique* **70**, 763-770 (2007).
23. Otsu N. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics* **9**, 62-66 (1979).
24. Gonzalez RC, Woods RE, Eddins SL. *Digital Image Processing Using MATLAB*. 2nd Edition, Gatesmark Publishing, USA (2009).
25. Saerens M, Latinne P, Decaestecker C. Any reasonable cost function can be used for a posteriori probability approximation. *IEEE Transactions on Neural Networks* **13**, 1204-1210 (2002).
26. Papadimitriou CH, Steiglitz K. *Combinatorial Optimization: Algorithms and Complexity*. Dover Publications, Mineola, NY, USA (1998).
27. Godinez WJ, Lampe M, Wörz S, Müller B, Eils R, Rohr K. Deterministic and probabilistic approaches for tracking virus particles in time-lapse fluorescence microscopy image sequences. *Medical Image Analysis* **13**, 325-342 (2009).
28. Godinez WJ, Lampe M, Eils R, Müller B, Rohr K. Tracking multiple particles in fluorescence microscopy images via probabilistic data association. *Proceedings of the IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 1925-1928 (2011).

29. Sage D, Neumann FR, Hediger F, Gasser SM, Unser M. Automatic tracking of individual fluorescence particles: Application to the study of chromosome dynamics. *IEEE Transactions on Image Processing* **14**, 1372-1383 (2005).
30. Marr D, Hildreth E. Theory of edge detection. *Proceedings of the Royal Society of London B: Biological Sciences* **207**, 187-217 (1980).
31. Blom HAP. An efficient filter for abruptly changing systems. *Proceedings of the IEEE Conference on Decision and Control*, 656-658 (1984).
32. Genovesio A, Liedl T, Emiliani V, Parak WJ, Coppey-Moisan M, Olivo-Marin J-C. Multiple particle tracking in 3-D+t microscopy: Method and application to the tracking of endocytosed quantum dots. *IEEE Transactions on Image Processing* **15**, 1062-1070 (2006).
33. Rink J, Ghigo E, Kalaidzidis Y, Zerial M. Rab conversion as a mechanism of progression from early to late endosomes. *Cell* **122**, 735-749 (2005).
34. Press WH, Teukolsky SA, Vetterling WT, Flannery BP. *Numerical Recipes in C: The Art of Scientific Computing*. 3rd Edition, Cambridge University Press, Cambridge, UK (2007).
35. Verestoy J, Chetverikov D, Marcell N. Digital particle image velocimetry: A challenge for feature based tracking. *Machine Graphics and Vision* **8**, 553-569 (1999).
36. Liang L, Shen H, De Camilli P, Duncan JS. Tracking clathrin coated pits with a multiple hypothesis based method. *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention* **6362**, 315-322 (2010).
37. Stallinga S, Rieger B. The effect of background on localization uncertainty in single emitter imaging. *Proceedings of the IEEE International Symposium on Biomedical Imaging*, 988-991 (2012).
38. Poore AB, Gadaleta S. Some assignment problems arising from multiple target tracking. *Mathematical and Computer Modeling* **43**, 1074-1091 (2006).
39. Li XR, Jilkov VP. Survey of maneuvering target tracking. Part V. Multiple-model methods. *IEEE Transactions on Aerospace and Electronic Systems* **41**, 1255-1321 (2005).
40. Li XR, Zhang YM. Numerically robust implementation of multiple-model algorithms. *IEEE Transactions on Aerospace and Electronic Systems* **36**, 266-278 (2000).
41. Yin Z, Kanade T, Chen M. Understanding the phase contrast optics to restore artifact-free microscopy images for segmentation. *Medical Image Analysis* **16**, 1047-1062 (2012).
42. Magnusson KEG, Jaldén J. A batch algorithm using iterative application of the Viterbi algorithm to track cells and construct cell lineages. *Proceedings of the IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 382-385 (2012).

43. Husain M, Boudier T, Paul-Gilloteaux P, Casuso I, Scheuring S. Software for drift compensation, particle tracking and particle analysis of high-speed atomic force microscopy image series. *Journal of Molecular Recognition* **25**, 292-298 (2012).
44. Casuso I, Khao J, Chami M, Paul-Gilloteaux P, Husain M, Duneau JP, Stahlberg H, Sturgis JN, Scheuring S. Characterization of the motion of membrane proteins using high-speed atomic force microscopy. *Nature Nanotechnology* **7**, 525-529 (2012).
45. Abràmoff MD, Magalhães PJ, Ram SJ. Image processing with ImageJ. *Biophotonics International* **11**, 36-42 (2004).
46. Racine V, Hertzog A, Jouanneau J, Salamero J, Kervrann C, Sibarita J-B. Multiple-target tracking of 3D fluorescent objects based on simulated annealing. *Proceedings of the IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 1020-1023 (2006).
47. Rao AR, Schunck BG. Computing oriented texture fields. *CVGIP: Graphical Models and Image Processing* **53**, 157-185 (1991).
48. Hager GD, Dewan M, Stewart CV. Multiple kernel tracking with SSD. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 790-797 (2004).
49. Rousseeuw PJ, Leroy AM. *Robust Regression and Outlier Detection*. Wiley, Hoboken, NJ, USA (1987).
50. Matusita K. Decision rules, based on the distance, for problems of fit, two samples, and estimation. *Annals of Mathematical Statistics* **26**, 631-640 (1955).
51. Thompson RE, Larson DR, Webb WW. Precise nanometer localization analysis for individual fluorescent probes. *Biophysical Journal* **82**, 2775-2783 (2002).
52. Shafique K, Shah M. A noniterative greedy algorithm for multiframe point correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**, 51-65 (2005).
53. Jaqaman K, Loerke D, Mettlen M, Kuwata H, Grinstein S, Schmid SL, Danuser G. Robust single-particle tracking in live-cell time-lapse sequences. *Nature Methods* **5**, 695-702 (2008).
54. Lowe DG. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* **60**, 91-110 (2004).
55. Munkres J. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics* **5**, 32-38 (1957).
56. Crocker C, Grier DG. Methods of digital video microscopy for colloidal studies. *Journal of Colloid and Interface Science* **179**, 298-310 (1996).

57. Celler K, van Wezel GP, Willemse J. Single particle tracking of dynamically localizing TatA complexes in *Streptomyces coelicolor*. *Biochemical and Biophysical Research Communications* **438**, 38-42 (2013).
58. Ku T-C, Huang Y-N, Huang C-C, Yang D-M, Kao L-S, Chiu T-Y, Hsieh C-F, Wu P-Y, Tsai Y-S, Lin C-C. An automated tracking system to measure the dynamic properties of vesicles in living cells. *Microscopy Research and Technique* **70**, 119-134 (2007).
59. Ku T-C, Kao L-S, Lin C-C, Tsai Y-S. Morphological filter improve the efficiency of automated tracking of secretory vesicles with various dynamic properties. *Microscopy Research and Technique* **72**, 639-649, (2009).
60. Li X, Hao M, Piston DW, Dawant BM. Automatic tracking of proteins in sequences of fluorescence images. *Proceedings of SPIE Medical Imaging: Image Processing* **5370**, 1364-1371 (2004).
61. Serra J. *Image Analysis and Mathematical Morphology*. Academic Press, London, UK (1982).

Supplementary Note 2: Simulated Image Data Sets

This document provides a detailed description of the image data used in this study to objectively evaluate the performance of the particle tracking methods. The actual data used in the training phase and the competition phase of the study can be downloaded from the website, as described below. The software developed to generate these data is described in **Supplementary Note 4**.

INTRODUCTION

An exact and objective evaluation of particle tracking methods requires a quantitative rather than a qualitative assessment of the tracking results. In turn, quantitative assessment requires the availability of the ground truth of the relevant particle properties captured by the image data. For real biological image data, this information is not available. At best, an estimate of the true particle positions and labels at each time point can be obtained from expert human observers, based on visual detection and manual annotation of the images. Since many biological studies today still involve some form of manual particle tracking,¹ it may seem sensible to compare the output of computational tracking methods with manually obtained reference data. However, there are at least three problems with this. First, the use of manual reference data reintroduces subjectivity, and requires dealing with intra- and inter-observer variability.^{2,3} Second, closely related to this, it has been demonstrated previously that human observers may easily produce inferior results compared to computational methods.³⁻⁶ Why compare with humans if we want computational methods to do a better job in the first place? Third, the high volume and content of the image data often far exceeds the capacity of human observers to track everything, forcing them to focus on only a limited subset of the data, thereby potentially introducing bias.⁶ For these reasons, the use of simulated test image data has become commonplace in the field.⁷⁻¹⁸ Not only does it allow for objective, consistent, and unbiased evaluations, it also allows to independently analyze, in a fully controlled fashion, the different factors affecting tracking performance. Therefore, similar to early evaluation studies,^{19,20} the present study was based on simulated image data.

SIMULATED IMAGE DATA

The appearance of particles in real biological image data is determined by many factors. These include the particular molecular complex studied, sample preparation (including labeling), microscope choice and settings, and detector (CCD or PMT) settings, all of which may potentially affect the performance of particle tracking methods. The space spanned by all imaging variables is very high-dimensional and

would correspondingly require a very large set of sample image data to investigate exhaustively. To make the study feasible, we focused on three key factors that in our experience have the largest influence on tracking performance, and we fixed other variables to commonly used, practical values. The three factors are: 1) particle dynamics (representing a biological scenario), 2) particle density (the number of particles in the field of view), and 3) particle signal (relative to noise). Here we provide further motivation for the choices made. The next section summarizes the image data set.

FACTOR 1: PARTICLE DYNAMICS

An important aspect of any particle tracking method is the ability to deal with a range of biological scenarios. Four different scenarios were simulated in this study, which we believe represent a wide variety of studies. Each scenario captured one particular type of particle dynamics and shape. In this study we focused on punctate objects (particles with dimensions much smaller than or on the order of the optical diffraction limit), which implies that shape is well modeled by the point-spread function (PSF) of the microscope. **Scenario 1** (nicknamed "Vesicles") simulated particles showing Brownian (random walk) motion, imaged in 2D+time using a widefield microscope. This is representative of normal diffusion, which occurs in many cellular processes.²¹⁻²³ **Scenario 2** ("Microtubules") simulated particles showing directed (near-constant velocity) motion, imaged in 2D+time using a confocal microscope (single plane mode). This is the only scenario where particles were modeled to be slightly elongated instead of spherical, with elongation in the direction of motion, representative of microtubule tips.²⁴⁻²⁶ **Scenario 3** ("Receptors") simulated particles switching between Brownian and directed motion models (with fixed probability given below), imaged in 2D+time using a confocal microscope (single plane mode). This type of dynamics may be observed with, for example, various types of receptor and motor proteins.^{21,27} Finally, **Scenario 4** ("Viruses") also simulated particles switching between Brownian and directed motion models, but moving roughly in the same direction, and imaged in 3D+time with a confocal microscope (stacks mode). This scenario is representative of, for example, the entry of viruses and their directed transport by motor proteins along distinct cytoskeletal tracks.^{28,29}

In this study, we simulated fluorescence microscopy imaging, using either a widefield or a confocal microscope as indicated above, since this is most commonly used in live-cell studies of intracellular dynamic processes.³⁰⁻³² This means that particles show as bright spots (relative to a darker background) as a result of incoherent light emitted by fluorescent labels. The imaging process can thus be accurately modeled as a convolution of the emitted light field with the diffraction-limited microscope PSF (assumed

to be space invariant).^{33,34} For widefield imaging, the classical model of a possibly defocused optical imaging system was used,³⁵ extended to include spherical aberrations due to refractive index mismatches.^{36,37} Numerical computation of the model was accomplished using the fast Fourier transform (FFT) and fixing the model parameters to the following values: numerical aperture of the objective lens $NA = 1.4$ (oil immersion), emission wavelength $\lambda = 520$ nm (green fluorescent light), refractive index of the immersion medium $n_i = 1.515$ (oil), and specimen refractive index $n_s = 1.33$ (watery substance). In the case of confocal imaging, the PSF is defined as the convolution of the widefield PSF with a pinhole model, the outcome of which we approximated by taking the squared magnitude of the widefield PSF model. This corresponds to using a very small pinhole and taking the excitation and emission wavelengths to be approximately equal. Alternatively, the confocal PSF can be approximated very accurately using Gaussian profiles.³⁸ With this in mind, and in order to simplify the calculations, we modeled the elongated particles in Scenario 2 directly by non-isotropic Gaussian profiles. Specifically, the basic model used was a 3D Gaussian profile with physical scales $\sigma_x = 0.21 \lambda / NA$ and $\sigma_z = 0.66 \lambda n_s / NA^2$ as used previously by others,³⁹ and with $\sigma_y = 4\sigma_x$. For each particle, at each time point, this profile was rotated in-plane to point in the direction of the velocity vector, the latter of which was initialized with zero z-component in the first time point of each image sequence. While in the other scenarios the pixel size was fixed to a practically realistic value of 67 nm, the pixel size in Scenario 2 was reduced to 50 nm, to "stretch" the particles to similar size (in pixels) as observed in real images of microtubule tips. The distance between optical planes in the 3D scenario was fixed to 300 nm.

The dynamics models of the particles in the different scenarios were implemented as follows. For every particle, going from the current time point to the next, Brownian motion was simulated by sampling from a normal (Gaussian) distribution centered at the current position and with standard deviation $\sigma_B = 2$ pixels in Scenario 1 and 0.6 pixels in Scenarios 3 and 4. Directed motion was simulated as near-constant velocity with small random accelerations to allow deviations from a straight path. Concretely, for every particle, the position in the next time point was computed as the current position plus the displacement dictated by the current velocity vector, and an additional probabilistic change in the position as well as in the velocity vector by drawing from a multivariate normal distribution with covariance matrix $\Sigma = q \cdot ((\sigma_{11}, \sigma_{12}), (\sigma_{21}, \sigma_{22}))$ for each coordinate.⁴⁰ The values of the parameters were set as follows: variance of the position noise $\sigma_{11} = 1/3$, covariance of position and velocity noise $\sigma_{12} = \sigma_{21} = 1/2$, variance of the velocity noise $\sigma_{22} = 1$, and influence factor $q = 0.6$ in Scenario 2 and 1.25 in Scenarios 3 and 4. During simulation the velocity was kept in (clipped to) the range from $v_{\min} = 3$ to $v_{\max} = 7$ pixels

per frame in Scenario 2 and from $v_{\min} = 2$ to $v_{\max} = 6$ pixels per frame in Scenarios 3 and 4. Finally, switching between (1) Brownian and (2) directed motion was modeled as a Markov chain with probability matrix $\mathbf{\Pi} = ((p_{11}, p_{12}), (p_{21}, p_{22}))$. The probabilities were set to the following values: $p_{11} = 0.90$, $p_{12} = 0.10$, $p_{21} = 0.15$, $p_{22} = 0.85$. In Scenario 4, in order to make all particles move roughly in the same direction, the velocity vector was reset to a fixed velocity vector after every switch from Brownian to directed motion. In all scenarios, motion was simulated in 3D, and imaged in either 2D (single plane) or 3D (stacks) over time as indicated, thus simulating the possibility of particles moving out of focus and changing appearance accordingly. All particles moved independently, that is particle interaction (splitting, merging, or any type of affection) was not modeled. The mentioned parameter values were chosen empirically to yield data visually mimicking real biological data from the different scenarios. Using these parameter values, and assuming a typical imaging rate of 1 frame per second (60 frames per minute), we observed the following mean particle velocities in the simulated data, which correlate well with measured velocities in experimental studies of real data (we cite example studies): $\sim 10 \mu\text{m}/\text{min}$ in Scenario 1 (certain membrane proteins⁴¹), $\sim 15 \mu\text{m}/\text{min}$ in Scenario 2 (typical microtubule growth rates²⁶), $\sim 7.5 \mu\text{m}/\text{min}$ in Scenario 3 (for example T-cell receptor dynamics⁴²), and $\sim 10 \mu\text{m}/\text{min}$ in Scenario 4 (various viruses⁴³). For the switching between Brownian and directed motion we used similar values for the probability matrix parameters as in a previous study.⁴⁴

FACTOR 2: PARTICLE DENSITY

The second factor that can be expected to have a large influence on tracking performance is the density of the particles in the scene: the higher the density, the higher the ambiguity of the data, as particles increasingly approach and cross each other, or even produce clutter. In this study, the field of view was fixed (to 512 by 512 pixels in-plane and with 10 planes in the 3D scenario), so that we may express density simply in terms of the number of particles. For each of the four biological scenarios described above, three levels of particle density were considered: **low density** ($N \sim 100$ particles), **mid density** ($N \sim 500$ particles), and **high density** ($N \sim 1000$ particles). For Scenario 2, which simulated somewhat larger particles (naturally resulting in a visually denser image), the numbers of particles were chosen a bit smaller, yielding more realistic data according to our expert biologists. To further increase the degree of realism, the exact number of particles in each case was governed by random processes. Specifically, starting with the target number of N particles distributed equally and independently over the field of view by drawing positions from a uniform distribution, particle disappearance was implemented as a Bernoulli process, with a fixed probability of $\alpha = 0.05$ for each particle to disappear at each time point.

In addition, particles could disappear simply by moving out of the field of view. In either case, the corresponding tracks were terminated after disappearance (the possibility of reappearance was excluded). At each time point, the number of newly appearing particles was sampled from a Poisson distribution, with mean equal to the target number of particles times the probability of disappearance: $\mu = \alpha \cdot N$. The positions of new particles were again drawn from a uniform distribution (independent of existing particles) and were allowed to be initially located outside the field of view (both laterally and axially) in order to simulate the entry of particles into the field of view. On average, particle disappearance and appearance compensated each other. In the end, of all generated tracks, only those lasting for at least 4 frames were retained. Due to the random (dis)appearance of particles, the total number of tracks generated per image sequence (with a fixed length of 100 frames) was about six times higher than the average number of particles per frame.

FACTOR 3: PARTICLE SIGNAL

The third aspect known from previous studies^{15,19,20} to have a major impact on particle detection and, consequently, on tracking performance, is the particle brightness or signal strength in the image relative to the background and the noise level. This experimental condition is conveniently summarized as the signal-to-noise ratio (SNR). Different definitions of SNR exist in the literature. While signal is commonly defined as the difference (contrast) between the peak object intensity I_o and the local mean background intensity level I_b , different choices are sometimes made regarding the noise. Since we are simulating fluorescence microscopy imaging, which is a photon counting process characterized by Poisson statistics, the noise is given by the square-root of intensity, which is (much) larger within a bright object as compared to the darker background. In contrast with other noise sources (thermal, readout, quantization), which can be reduced to be negligible in practice, photon noise is unavoidable and is the dominant noise source to be modeled.^{45,46} Following earlier studies,^{7,15,19} we therefore employed the definition $\text{SNR} = (I_o - I_b) / \sqrt{I_o}$, and we set the mean background level I_b to a value in the range of 10 to 20 photons per pixel (a randomly selected and fixed value per data set) to represent detector noise. For each scenario, and for each density level, image sequences were generated at four SNR levels. Since **SNR = 4** is known to be a critical level at which various popular methods start to break down,^{15,19} we selected this level and added one considerably higher level of **SNR = 7** that should be relatively easy for all methods, one considerably lower level of **SNR = 2** that is representative of many practical live-cell imaging experiments, and an extremely low level of **SNR = 1** to challenge even the best algorithms. Since

particles may move out of focus and thus lose intensity, a given SNR level indicates the maximum possible SNR, corresponding to particles that are in-focus.

IMAGE DATA SETS

The four scenarios, times three density levels, times four SNR levels, amount to a data set consisting of 48 image sequences, totaling 4 GB of data (with quantization of 8 bits/pixel in all cases). The software developed for this study (see **Supplementary Note 4**) is able to automatically generate all 48 image sequences for a user-given choice of the seed value initializing the various random processes involved: picking the initial positions of the particles, computing the new positions and (in the case of directed motion) the velocities of the particles at each time point, switching (where applicable) between Brownian and directed motion, determining whether to terminate or continue a particle track, how many and where new particle tracks are to be initialized, and the image noise. Each seed value yields a unique instance of the image data set, which is totally different from any other instance in terms of the details, but equivalent in terms of statistics. In the training phase of the study, participants were given one instance of the data set, together with the basic information listed in **Table 2** of the main paper, and the ground truth particle positions to train, improve, and test their algorithms. In the actual competition phase of the study, participants were given another instance of the data set, without the ground truth, and were asked to submit their tracking results to the independent evaluator (M.M.), who was not a contestant and the only one to know the ground truth. An impression of the image data (from the training phase of the study) can be obtained from **Figure 1** of the main paper. Corresponding videos of the sample images in that figure are shown in **Supplementary Videos 1-10**.

The actual data used in the training and in the competition phase can be downloaded here:

URL: <http://bioimageanalysis.org/track/>

Login: anonymous@bioimageanalysis.org

Password: erfg14d3

REFERENCES

1. Meijering E, Dzyubachyk O, Smal I. Methods for cell and particle tracking. *Methods in Enzymology* **504**, 183-200 (2012).

2. Webb D, Hamilton MA, Harkin GJ, Lawrence S, Camper AK, Lewandowski Z. Assessing technician effects when extracting quantities from microscope images. *Journal of Microbiological Methods* **53**, 97-106 (2003).
3. Bahnson A, Athanassiou C, Koebler D, Qian L, Shun T, Shields D, Yu H, Wang H, Goff J, Cheng T, Houck R, Cowser L. Automated measurement of cell motility and proliferation. *BMC Cell Biology* **6**, 19 (2005).
4. Dorn F, Danuser G, Yang G. Computational processing and analysis of dynamic fluorescence image data. *Methods in Cell Biology* **85**, 497-538 (2008).
5. Smal I, Draegestein K, Galjart N, Niessen W, Meijering E. Particle filtering for multiple object tracking in dynamic fluorescence microscopy images: Application to microtubule growth analysis. *IEEE Transactions on Medical Imaging* **27**, 789-804 (2008).
6. Huth J, Buchholz M, Kraus JM, Schmucker M, von Wichert G, Krndija D, Seufferlein T, Gress TM, Kestler HA. Significantly improved precision of cell migration analysis in time-lapse video microscopy through use of a fully automated tracking system. *BMC Cell Biology* **11**, 24 (2010).
7. Sbalzarini IF, Koumoutsakos P. Feature point tracking and trajectory analysis for video imaging in cell biology. *Journal of Structural Biology* **151**, 182-195 (2005).
8. Genovesio A, Liedl T, Emiliani V, Parak WJ, Coppey-Moisan M, Olivo-Marin J-C. Multiple particle tracking in 3-D+t microscopy: Method and application to the tracking of endocytosed quantum dots. *IEEE Transactions on Image Processing* **15**, 1062-1070 (2006).
9. Zhao T, Murphy RF. Automated learning of generative models for subcellular location: Building blocks for systems biology. *Cytometry Part A* **71**, 978-990 (2007).
10. Yoon JW, Bruckbauer A, Fitzgerald WJ, Klenerman D. Bayesian inference for improved single molecule fluorescence tracking. *Biophysical Journal* **94**, 4932-4947 (2008).
11. Jaqaman K, Loerke D, Mettlen M, Kuwata H, Grinstein S, Schmid SL, Danuser G. Robust single-particle tracking in live-cell time-lapse sequences. *Nature Methods* **5**, 695-702 (2008).
12. Lehmussola A, Ruusuvaari P, Selinummi J, Rajala T, Yli-Harja O. Synthetic images of high-throughput microscopy for validation of image analysis methods. *Proceedings of the IEEE* **96**, 1348-1360 (2008).
13. Godinez WJ, Lampe M, Wörz S, Müller B, Eils R, Rohr K. Deterministic and probabilistic approaches for tracking virus particles in time-lapse fluorescence microscopy image sequences. *Medical Image Analysis* **13**, 325-342 (2009).
14. Svoboda D, Kozubek M, Stejskal S. Generation of digital phantoms of cell nuclei and simulation of image formation in 3D image cytometry. *Cytometry Part A* **75**, 494-509 (2009).

15. Smal I, Loog M, Niessen W, Meijering E. Quantitative comparison of spot detection methods in fluorescence microscopy. *IEEE Transactions on Medical Imaging* **29**, 282-301 (2010).
16. Feng L, Xu Y, Yang Y, Zheng X. Multiple dense particle tracking in fluorescence microscopy images based on multidimensional assignment. *Journal of Structural Biology* **173**, 219-228 (2011).
17. Smith MB, Karatekin E, Gohlke A, Mizuno H, Watanabe N, Vavylonis D. Interactive, computer-assisted tracking of speckle trajectories in fluorescence microscopy: Application to actin polymerization and membrane fusion. *Biophysical Journal* **101**, 1794-1804 (2011).
18. Matov A, Edvall MM, Yang G, Danuser G. Optimal-flow minimum-cost correspondence assignment in particle flow tracking. *Computer Vision and Image Understanding* **115**, 531-540 (2011).
19. Cheezum MK, Walker WF, Guilford WH. Quantitative comparison of algorithms for tracking single fluorescent particles. *Biophysical Journal* **81**, 2378-2388 (2001).
20. Carter BC, Shubeita GT, Gross SP. Tracking single particles: A user-friendly quantitative evaluation. *Physical Biology* **2**, 60-72 (2005).
21. Saxton MJ, Jacobson K. Single-particle tracking: Applications to membrane dynamics. *Annual Review of Biophysics and Biomolecular Structure* **26**, 373-399 (1997).
22. Ruprecht V, Axmann M, Wieser S, Schutz GJ. What can we learn from single molecule trajectories? *Current Protein & Peptide Science* **12**, 714-724 (2011).
23. Jandt U, Zeng A-P. Modeling of intracellular transport and compartmentation. *Advances in Biochemical Engineering/Biotechnology* **127**, 221-249 (2012).
24. Galjart N. Plus-end-tracking proteins and their interactions at microtubule ends. *Current Biology* **20**, 528-537 (2010).
25. Akhmanova A, Steinmetz MO. Microtubule +TIPs at a glance. *Journal of Cell Science* **123**, 3415-3419, (2010).
26. Matov A, Applegate K, Kumar P, Thoma C, Krek W, Danuser G, Wittmann T. Analysis of microtubule dynamic instability using a plus-end growth marker. *Nature Methods* **7**, 761-768 (2010).
27. Fujita K, Iwaki M, Iwane AH, Marcucci L, Yanagida T. Switching of myosin-V motion between the lever-arm swing and Brownian search-and-catch. *Nature Communications* **3**, 956 (2012).
28. Helmuth JA, Burckhardt CJ, Koumoutsakos P, Greber UF, Sbalzarini IF. A novel supervised trajectory segmentation algorithm identifies distinct types of human adenovirus motion in host cells. *Journal of Structural Biology* **159**, 347-358 (2007).
29. Ewers H, Schelhaas M. Analysis of virus entry and cellular membrane dynamics by single particle tracking. *Methods in Enzymology* **506**, 63-80 (2012).

30. Stephens DJ, Allan VJ. Light microscopy techniques for live cell imaging. *Science* **300**, 82-86 (2003).
31. Lichtman JW, Conchello J-A. Fluorescence microscopy. *Nature Methods* **2**, 910-919 (2005).
32. Vonesch C, Aguet F, Vonesch J-L, Unser M. The colored revolution of bioimaging. *IEEE Signal Processing Magazine* **23**, 20-31 (2006).
33. Born M, Wolf E. *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light*. 7th edition, Cambridge University Press, Cambridge, UK (1999).
34. Gu M. *Advanced Optical Imaging Theory*. Springer-Verlag, Berlin, Germany (2000).
35. Stokseth PA. Properties of a defocused optical system. *Journal of the Optical Society of America* **59**, 1314-1321 (1969).
36. Gibson SF, Lanni F. Experimental test of an analytical model of aberration in an oil-immersion objective lens used in three-dimensional light microscopy. *Journal of the Optical Society of America A: Optics and Image Science* **8**, 1601-1613 (1991).
37. Pankajakshan P, Blanc-Féraud L, Kam Z, Zerubia J. Point-spread function retrieval for fluorescence microscopy. *Proceedings of the IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 1095-1098 (2009).
38. Zhang B, Zerubia J, Olivo-Marin J-C. Gaussian approximations of fluorescence microscope point-spread function models. *Applied Optics* **46**, 1819-1829 (2007).
39. Thomann D, Rines DR, Sorger PK, Danuser G. Automatic fluorescent tag detection in 3D with super-resolution: Application to the analysis of chromosome movement. *Journal of Microscopy* **208**, 49-64 (2002).
40. Bar-Shalom Y, Li XR, Kirubarajan T. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, New York, NY, USA (2001).
41. Yang L, Dun AR, Martin KJ, Qiu Z, Dunn A, Lord GJ, Lu W, Duncan RR, Rickman C. Secretory vesicles are preferentially targeted to areas of low molecular SNARE density. *PLoS One* **7**, e49514 (2012).
42. Moss WC, Irvine DJ, Davis MM, Krummel MF. Quantifying signaling-induced reorientation of T cell receptors during immunological synapse formation. *Proceedings of the National Academy of Sciences of the United States of America* **99**, 15024-15029 (2002).
43. Arhel N, Genovesio A, Kim K-A, Miko S, Perret E, Olivo-Marin J-C, Shorte S, Charneau P. Quantitative four-dimensional tracking of cytoplasmic and nuclear HIV-1 complexes. *Nature Methods* **3**, 817-824 (2006).

44. Smal I, Meijering E, Draegestein K, Galjart N, Grigoriev I, Akhmanova A, van Royen ME, Houtsmuller AB, Niessen W. Multiple object tracking in molecular bioimaging by Rao-Blackwellized marginal particle filtering. *Medical Image Analysis* **12**, 764-777 (2008).
45. van Vliet LJ, Sudar D, Young IT. Digital fluorescence imaging using cooled CCD array cameras. In *Cell Biology: A Laboratory Handbook*, Celis JE (ed.), 2nd Edition, volume III, Academic Press, New York, NY, USA, 109-120 (1998).
46. Sheppard CJR, Gan X, Gu M, Roy M. Signal-to-noise ratio in confocal microscopes. In *Handbook of Biological Confocal Microscopy*, Pawley JB (ed.), 3rd Edition, Springer, New York, NY, Chapter 22, 442-452 (2006).

Supplementary Note 3: Performance Measures

This note provides a detailed description of the measures used in this study to quantitatively and objectively evaluate the performance of the particle tracking methods. The software developed to compute these performance measures is described in **Supplementary Note 4**.

INTRODUCTION

The problem of performance evaluation of tracking methods occurs in many fields, including computer vision, aerospace applications (radar tracking, navigation, traffic control), and biomedical research. Despite much consideration in the past decades,¹⁻³ there is as yet no single, well-accepted method to evaluate overall tracking performance. This can be explained by the fact that different application areas may be concerned with different aspects of track estimation and, consequently, may require different performance measures. In particular, measures proposed in other fields are often not applicable to biological particle tracking, where one is faced with a priori unknown and varying numbers of particles, whose identities are to be preserved throughout the image sequence.^{4,5}

A key aspect of comparing a set of estimated objects to a set of known but possibly a different number of ground-truth objects, is the pairing of their elements: which element in the former should be compared to which element in the latter? A sensible approach to solving this problem for sets of positions is the use of optimal subpattern assignment.^{6,7} This concept has recently also been extended to sets of labeled object tracks.⁸ The evaluation of particle tracking methods in the present study was based on the same underlying idea, as described in detail below. Nevertheless, in order to have a complete and intuitive characterization of the performance of the different methods, a set of complementary performance measures was used, rather than a single measure.

TRACK DEFINITION

A track is a temporal series of subsequent spatial positions. The spatial position at a given time point $t \geq 0$ is a vector $\theta(t) = (x(t), y(t), z(t))$, with $x(t)$, $y(t)$, and $z(t)$ the coordinates at this time along the respective axes of the image. In a 3D image sequence, all three coordinates may vary, while in a 2D image sequence, the z coordinate is fixed. A track θ existing from time $t_{\text{init}} \geq 0$ to time $t_{\text{end}} \geq t_{\text{init}}$ is therefore defined as the set $\theta = \{\theta(t), t = t_{\text{init}}, \dots, t_{\text{end}}\}$. Missing positions in the interval $[t_{\text{init}}, t_{\text{end}}]$ are marked as non-matching and are penalized as described below.

DISTANCE BETWEEN TWO TRACKS

For the purpose of measuring the distance between two tracks, the following gated Euclidean distance between two positions $\theta_1(t)$ and $\theta_2(t)$ is defined:

$$\|\theta_1(t) - \theta_2(t)\|_{2,\epsilon} \triangleq \min(\|\theta_1(t) - \theta_2(t)\|_2, \epsilon)$$

with $\|\cdot\|_2$ the standard l_2 norm of \mathbb{R}_3 , and $\epsilon \in \mathbb{R}_+$ the gate. The rationale behind the use of the gate ϵ is to limit the penalization of tracks that separate. When two tracks are more than ϵ apart at any time t , it is indeed considered that their positions do not match at that time point. In that case, it is irrelevant to measure the actual distance between these positions, and a fixed penalty ϵ is used instead. In the context of this study, the value of ϵ was set to 5 pixels, which, for the imaging parameters simulated in our data (see **Supplementary Note 2**), was on the order of the Rayleigh criterion.^{9,10} In other words, the required minimum distance between diffraction-limited particles to allow visual separation (Rayleigh), was taken as the maximum tolerable distance for the particle tracking methods.

It may happen that two tracks have different temporal supports. For instance, θ_1 may exist at a given time t , while θ_2 does not. In that case, we consider that the tracks do not match at that time point, and the distance between the tracks is defined to be equal to the penalty ϵ . If neither of the two tracks exist at time t , their distance is defined to be 0. This allows for the following compact formulation of the distance d between any two tracks θ_1 and θ_2 :

$$d(\theta_1, \theta_2) \triangleq \sum_{t=0}^{T-1} \|\theta_1(t) - \theta_2(t)\|_{2,\epsilon}$$

where T is the length (the number of frames) of the image sequence.

DISTANCE BETWEEN TWO TRACK SETS

Let $X = \{\theta_k^X, k = 1, \dots, |X|\}$ be an ordered set of ground-truth tracks, and Y a set of estimated tracks, whose similarity to X needs to be evaluated. Since some tracks in Y may not match a track in X , or vice versa, Y is extended with $|X|$ dummy tracks that are empty. Let \tilde{Y} denote this extended set of estimated tracks. Furthermore, let Ω be the ensemble of ordered sets of tracks that can be obtained by taking $|X|$ elements from \tilde{Y} . The distance between any $Z \in \Omega$ and X is then defined as the sum of the distances

between the $|X|$ pairs of tracks given by the ordering of the two sets. This allows for the definition of the distance between X and Y as the minimum distance between X and all possible Z :

$$d(X, Y) \triangleq \min_{Z \in \Omega} \sum_{k=1}^{|X|} d(\theta_k^X, \theta_k^Z)$$

Building the set of tracks $Z^* \in \Omega$ that minimizes the distance to X , involves reordering \tilde{Y} and taking a subset of $|X|$ elements from it. This task can be viewed as a rectangular assignment problem between the tracks in X and \tilde{Y} . Because of the additivity and positivity of the cost of track association according to the above definition, this problem can be solved in polynomial time, using the Munkres algorithm.¹¹

PERFORMANCE MEASURES

In order to evaluate the performance of any particle tracking method for any data, the output track set Y of the method was scored with respect to the ground-truth track set X of that data using the following measures, based on the optimal pairing (X, Z^*) described above:

- 1) The measure $\alpha(X, Y) \triangleq 1 - d(X, Y)/d(X, \emptyset)$, where \emptyset denotes the set of $|X|$ dummy tracks. By definition of $\|\cdot\|_{2,\epsilon}$, the lower bound of $d(X, Y)$ is 0, and the upper bound is $d(X, \emptyset)$. Indeed, a pair of tracks (θ_i^X, θ_j^Y) is guaranteed not to be selected by the optimization process if the distance between them is larger than the distance between θ_i^X and a dummy track. The value of measure $\alpha(X, Y)$ therefore lies in the interval $[0, 1]$. It takes value 1 if the pairs of tracks in X and Z^* match exactly (the distance between each pair of tracks is 0). It takes value 0 if no valid match could be found, that is if $Z^* = \emptyset$. It scores the best possible pairing of tracks between X and Y , and ignores the tracks in Y that did not make it into Z^* .
- 2) The measure $\beta(X, Y) \triangleq (d(X, \emptyset) - d(X, Y))/(d(X, \emptyset) + d(\bar{Y}, \emptyset))$, where $\bar{Y} = (\tilde{Y} \setminus Z^*) \cap Y$ denotes the set of tracks in Y that did not make it into Z^* , and \emptyset contains the appropriate number of dummy tracks, being $|X|$ for $d(X, \emptyset)$ and $|\bar{Y}|$ for $d(\bar{Y}, \emptyset)$. A track in Y may not have been selected for Z^* because either another estimated track or a dummy track was preferred over it. Such a spurious track typically consists of a combination of positions corresponding to different ground-truth tracks or to erroneous positions originating from clutter. The value of $\beta(X, Y)$ lies in $[0, \alpha(X, Y)]$. It takes value $\alpha(X, Y)$ if there are no spurious tracks in Y , that is if $\bar{Y} = \emptyset$. And it converges to 0 as the number of spurious tracks increases.

The measures α and β account for both association errors and localization errors. For a more detailed analysis, it is useful to separate these two types of errors. To evaluate association performance, the positions at time t of two paired tracks, $\theta_k^X(t)$ and $\theta_k^{Z^*}(t)$, are counted as matching if they are both non-dummy and $\|\theta_k^X(t) - \theta_k^{Z^*}(t)\|_2 < \epsilon$. Otherwise, they are counted as non-matching, with the exception of two dummy positions, which are not counted. This leads to the following measures:

- 3) The number of matching positions of the optimal pairs of tracks (X, Z^*) . These are referred to as true positive (TP) position pairs.
- 4) The number of positions in X that are paired with a dummy position in Z^* . These are referred to as false negative (FN) position pairs because the dummy positions *are* nevertheless associated with track positions in the ground-truth set X .
- 5) The number of positions in the spurious tracks \bar{Y} and the non-matching positions in Z^* . These are referred to as false positive (FP) positions because they correspond to estimated positions that were not associated with track positions in the ground-truth set X .
- 6) The Jaccard similarity coefficient¹² for positions, defined as $\text{JSC} \triangleq \text{TP}/(\text{TP} + \text{FN} + \text{FP})$, which lies in the interval $[0,1]$. It takes value 1 only if all position pairs in (X, Z^*) are matching and $\bar{Y} = \emptyset$. It converges to 0 as the number of non-matching pairs and/or positions in \bar{Y} increases.

It may also be useful to evaluate the association performance at the track level, rather than the position level. This leads to the following measures, analogous to the previous four measures:

- 7) The number of non-dummy tracks in Z^* . They are referred to as true positive tracks (TP_θ) because each of them contains a majority of matching positions with a single associated track in X .
- 8) The number of dummy tracks in Z^* . These are referred to as false negative tracks (FN_θ) because each of them *is* nevertheless associated with a single track in X .
- 9) The number of tracks in \bar{Y} . These are referred to as false positive tracks (FP_θ) because none of them is associated with a track in X .
- 10) The Jaccard similarity coefficient for tracks, defined as $\text{JSC}_\theta \triangleq \text{TP}_\theta/(\text{TP}_\theta + \text{FN}_\theta + \text{FP}_\theta)$, which lies in the interval $[0,1]$. It takes value 1 only if Z^* does not contain dummy tracks and $\bar{Y} = \emptyset$. It converges to 0 as the number of dummy tracks in Z^* and/or tracks in \bar{Y} increases.

Finally, the localization performance is characterized by the Euclidean distance (referred to as the error) between the positions of paired tracks. Since non-matching positions are already penalized by the above measures, the computation of localization errors is limited to matching positions:

- 11) The root mean-square error (RMSE) in true positive position pairs (TP as above).
- 12) The minimum error (Min) in TP position pairs.
- 13) The maximum error (Max) in TP position pairs.
- 14) The standard deviation (SD) of error in TP position pairs.

EXAMPLE CASES AND PERFORMANCE VALUES

To illustrate the effect of various tracking errors on the different performance measures, we provide several synthetic examples of increasing complexity. In the examples, a track is graphically represented as a series of point markers whose centers indicate the spatial position of the underlying particle at different time points, which are projected into a single image (**Figure N1**). The progression through time is indicated by a line connecting the point markers of the track. Tracks from the ground-truth set X are indicated by square-shaped markers connected by solid lines, while tracks from the estimated set Y are indicated by cross-shaped markers connected by dotted lines.

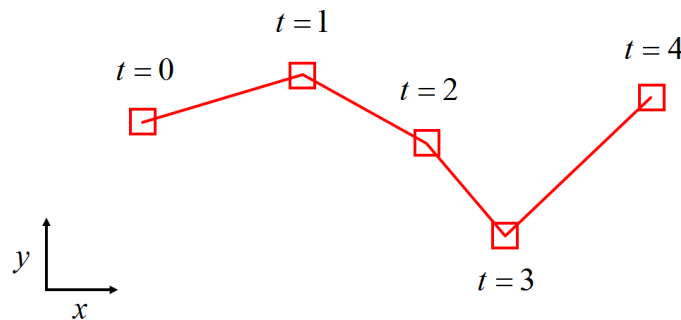


Figure N1: Ground-truth track defined for five successive time points ($t = 0, \dots, 4$). In the sequel we will omit the time labels from the point markers and consider the left-most marker as the starting point of the track.

CASE 1: NO ESTIMATED TRACKS

We start with the pathological case in which we have a single particle with ground truth (X as given in **Figure N1**) and the particle tracking method could not find any part of the track ($Y = \emptyset$). In this case, by definition, $\alpha = \beta = 0$. Also, since there are no estimated tracks at all, we have $TP_\theta = FP_\theta = 0$, and instead a dummy track is paired with the ground-truth track, yielding $FN_\theta = 1$, leading to $JSC_\theta = 0$. Similarly, we have $TP = FP = 0$, and since the ground-truth track covers five time points, we have the same number of matching dummy positions, $FN = 5$, yielding $JSC = 0$. Without any TP positions it is not possible to assess the localization performance of the tracking method.

CASE 2: ESTIMATED TRACKS IDENTICAL TO GROUND-TRUTH TRACKS

The other extreme is the case where the output of the particle tracking method is identical to the ground truth ($Y = X$ as in **Figure N2**). In this case, the number of matching tracks in Y is exactly the number of elements in X , here $TP_\theta = 1$, and since there are no dummy or spurious tracks, we have $FN_\theta = FP_\theta = 0$, and thus $JSC_\theta = 1$. Similarly, the number of positions in Y is exactly the number of positions in X , here $TP = 5$, and $FN = FP = 0$, yielding $JSC = 1$. Because the distance between each pair of estimated and ground-truth positions is 0, we have $\alpha = \beta = 1$, and all localization errors are 0.

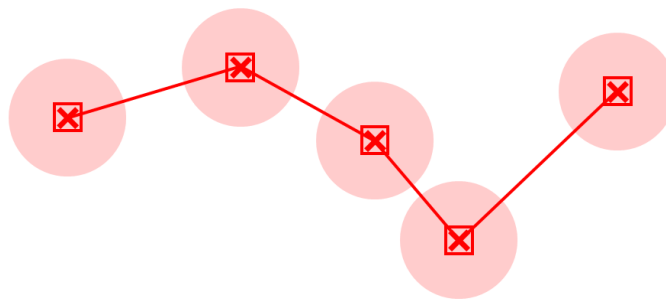


Figure N2: Ground-truth track (square-shaped markers connected by a solid line) with a perfectly matching estimated track (cross-shaped markers connected by a dotted line that is fully overlapping and thus not visible). The larger, light colored circles around the ground-truth positions indicate the gate ϵ within which estimated positions are searched.

CASE 3: FULLY MATCHING BUT NOT IDENTICAL TRACKS

In this example, we consider a similar situation as in Case 2, where an optimal pairing of estimated and ground-truth tracks is possible without the need for dummy tracks and without leaving spurious tracks, but where the estimated positions are not identical to the ground-truth positions, although they are within the gates of the latter (**Figure N3**). The distortions in the estimated positions affect only α and β and the localization measures. Since there are no spurious tracks, $\beta = \alpha$, and in this example both drop to 0.364. The localization errors all become > 0 (we refer to **Table N1** at the end of this section for the values of the performance measures for all example cases discussed).

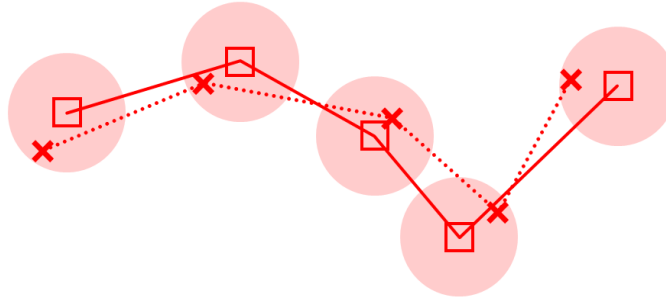


Figure N3: Ground-truth track (squares connected by solid lines) with a paired estimated track (crosses connected by dotted lines) whose positions are not identical with, but fall within, the gates (light colored circles) of the ground-truth positions.

CASE 4: MATCHING TRACKS HAVING NON-MATCHING POSITIONS

Next, we reconsider the situation of Case 3, and move one of the estimated track positions out of the gate of the corresponding ground-truth track position (**Figure N4**).

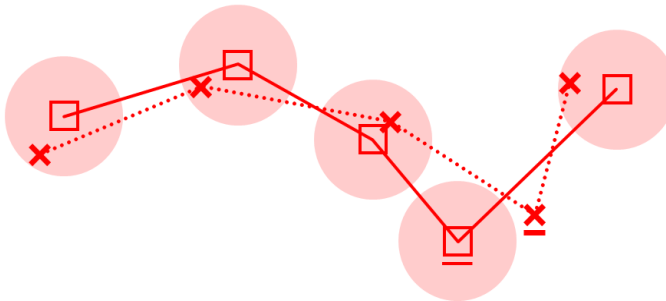


Figure N4: Estimated track (crosses connected by dotted lines) that is paired with a ground-truth track (squares connected by solid lines) but with one position (underscored cross) falling outside the gate of the ground-truth track position (underscored square).

As a result, this estimated position is considered non-matching with the ground truth, which translates into an increase in the number of false-positive positions, $FP = 1$. At the same time, the ground-truth position is now matched with a dummy position, leading to $FN = 1$, and since $TP = 4$, we have $JSC = 0.667$. The dummy position receives a penalty that is larger than the localization error of the original position in Case 3, leading to a decrease of both α and β . Also, the computation of RMSE, which is limited to TP positions only, no longer includes the now non-matching position, whose localization

error was relatively large, and as a result RMSE slightly decreases while SD slightly increases. All other performance measures remain unaffected by the change (**Table N1**).

CASE 5: MATCHING TRACKS WITH BIRTH AND DEATH MISMATCHES

In the previous examples we compared tracks (estimated versus ground truth) that were defined over the same time interval. Now we consider a case in which the existence window of the estimated track is shifted by one time point compared to the ground-truth track (**Figure N5**). This is representative of cases where a particle tracking method fails to detect the right birth and death times of a particle.

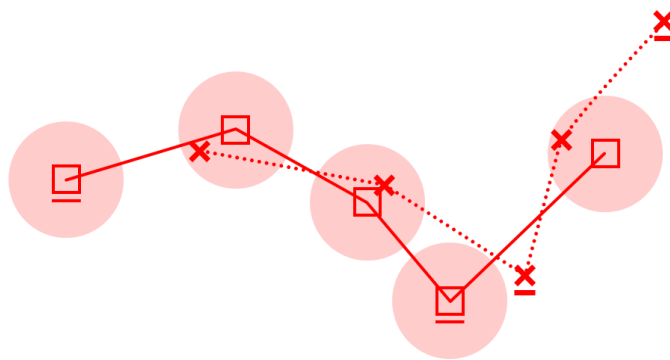


Figure N5: Estimated track (crosses connected by dotted lines) that is paired with a ground-truth track (squares connected by solid lines) but whose start and end positions do not correspond to the ground truth (all problematic positions are underscored).

Since the majority of the track positions still match, the two tracks are paired by the track association algorithm, and similar to Case 4 only the performance measures accounting for position matching and localization errors are affected (**Table N1**). Specifically, since TP further decreases to 3, and FP increases to 2, and the number of matched dummy positions FN also increases to 2, JSC drops to 0.429, and both α and β significantly drop to 0.052. Because we are now missing one more position (the left-most) with a relatively large localization error, again RMSE slightly decreases and SD slightly increases.

CASE 6: MULTIPLE ESTIMATED AND GROUND-TRUTH TRACKS

In the following examples (Cases 6-10) we consider multiple estimated and ground-truth tracks. Together they cover all the types of situations encountered in our study. As a first example we extend Case 5 with one additional estimated and corresponding ground-truth track (**Figure N6**).

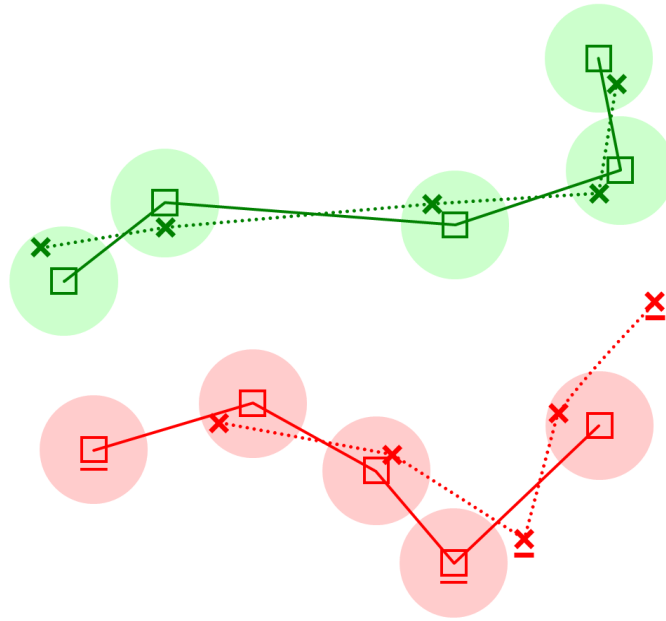


Figure N6: Multiple estimated tracks (crosses connected by dotted lines) that are paired with corresponding ground-truth tracks (squares connected by solid lines). The pairing produced by the track association algorithm is indicated by the coloring (the red estimated track is paired with the red ground-truth track, and similar for the green tracks).

Since the newly added estimated track and ground-truth track match both spatially and temporally, with only small inaccuracies in the estimated positions, they are correctly paired by the track association algorithm. As a result, similar to Case 5, only the performance measures accounting for position matching and localization errors are affected (**Table N1**). In particular, TP now increases to 8, making JSC increase again to 0.667, and also α and β increase significantly, to 0.256. Because the newly added estimated track has relatively low localization errors, both RMSE and SD decrease.

CASE 7: MISSING ESTIMATED TRACKS

In this example we revisit Case 6 and consider the situation in which there are two ground-truth tracks but only one estimated track (**Figure N7**). This is representative of cases where a particle tracking method produces less tracks than the number of ground-truth tracks. The estimated track is paired with one of the ground-truth tracks (the one whose positions match best) while the remaining ground-truth track is matched with a dummy track. As a result, $TP_{\theta} = FN_{\theta} = 1$, and since still $FP_{\theta} = 0$, we have $JSC_{\theta} = 0.5$. Because the number of TP positions decreases back to the level of Case 5, while the number of FN positions increases by the same amount, the value of JSC now drops significantly, as do the values

of both α and β (**Table N1**). And since localization errors are computed only for TP positions, the values of the corresponding performance measures are the same as in Case 5.

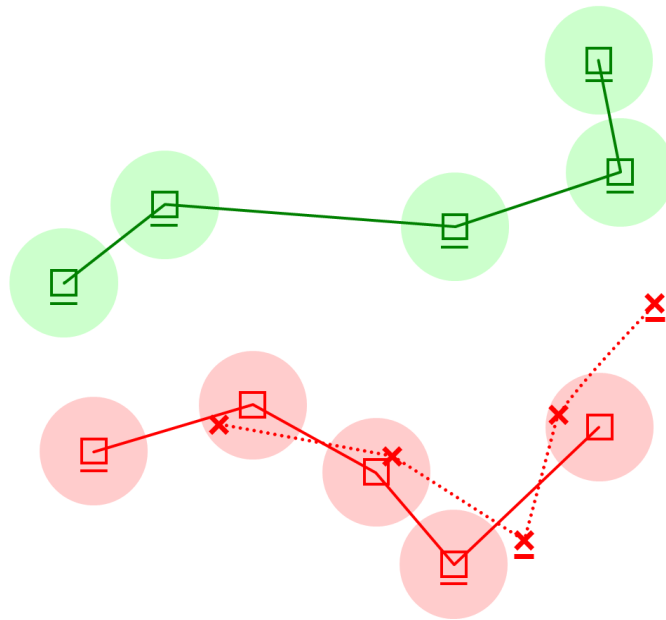
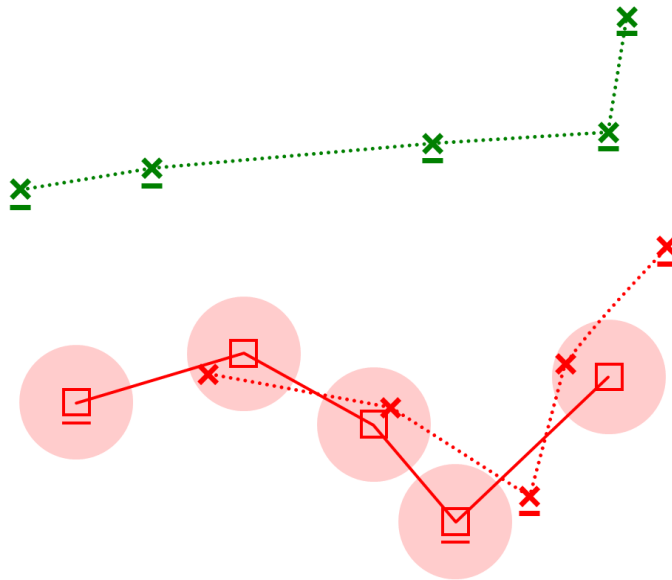


Figure N7: Two ground-truth tracks (squares connected by solid lines) but only one estimated track (crosses connected by dotted lines). The pairing produced by the track association algorithm is indicated by the coloring (the red estimated track is paired with the red ground-truth track while the green ground-truth track is paired with a dummy).

CASE 8: SPURIOUS ESTIMATED TRACKS

Here we again revisit Case 6, but consider the reverse situation as in Case 7, in that we now have two estimated tracks but only one ground-truth track (**Figure N8**). This is representative of cases where a particle tracking method produces more tracks than the number of ground-truth tracks. One of the estimated tracks (whose positions match best) is paired with the single ground-truth track while the other estimated track is considered spurious and remains non-paired. As a result, $TP_\theta = FP_\theta = 1$, and since $FN_\theta = 0$, we still have $JSC_\theta = 0.5$. Because the FN and FP positions are interchanged compared to Case 7, the value of JSC remains the same, but α increases back to the level of Case 5 (**Table N1**), as it does not penalize spurious tracks. And since localization errors are computed only for TP positions, the values of the corresponding performance measures are the same as in Cases 5 and 7.



CASE 9: BROKEN ESTIMATED TRACKS

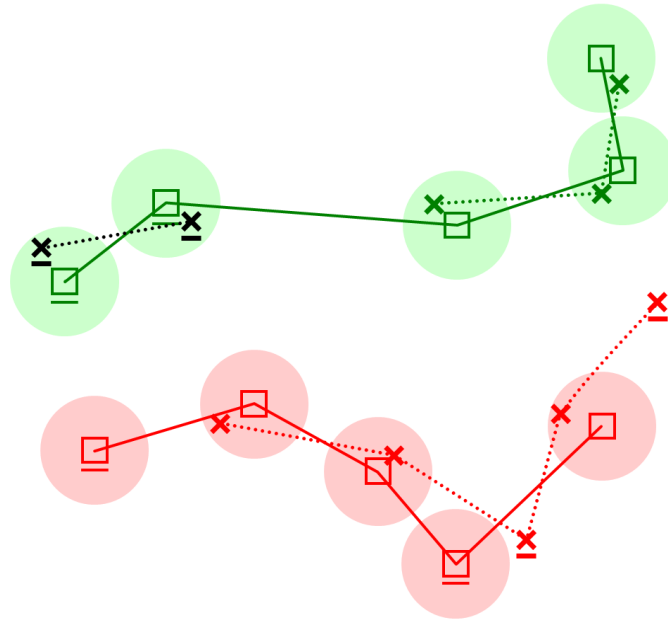


Figure N9: Two ground-truth tracks (squares connected by solid lines) but three estimated tracks (crosses connected by dotted lines) resulting from a linking failure. The pairing produced by the track association algorithm is indicated by the coloring (the red estimated track is paired with the red ground-truth track, and the green estimated track is paired with the green ground-truth track, while the black estimated track is considered spurious).

CASE 10: MIXED UP ESTIMATED TRACKS

In this last example we revisit Case 9 and consider estimated tracks consisting of detected particle positions belonging to different ground-truth tracks (**Figure N10**). This is representative of cases where a particle tracking method erroneously switches particle tracks. In the particular case considered here, one estimated track is paired with one of the ground-truth tracks, thus $TP_{\theta} = 1$, but the other estimated track does not match with the other ground-truth track, as the majority of its positions is too far off. It thus remains non-paired, yielding $FP_{\theta} = 1$, and consequently the ground-truth track is paired with a dummy track, yielding $FN_{\theta} = 1$, and thus $JSC_{\theta} = 0.333$. In terms of positions we find that $FP = FN = 7$, while $TP = 3$, resulting in $JSC = 0.176$. Both α and β decrease compared to Case 9 (**Table N1**) while remaining different from each other due to the presence of a spurious track. Finally, the localization errors are now computed based on only three TP positions, and are relatively small.

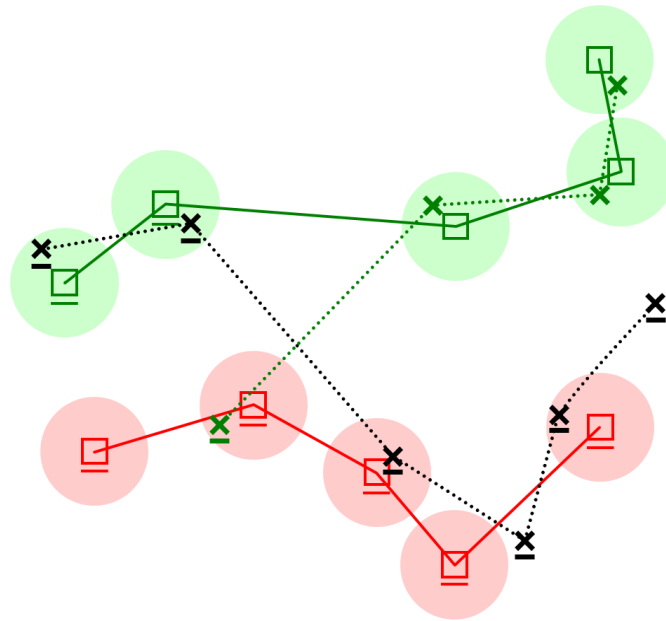


Figure N10: Two ground-truth tracks (squares connected by solid lines) and two estimated tracks (crosses connected by dotted lines) resulting from linking failures. The pairing produced by the track association algorithm is indicated by the coloring (the green estimated track is paired with the green ground-truth track while the black estimated track is not paired but considered spurious).

Table N1: Overview of the performance values for all discussed example cases. All floating-point values are given with three decimal places.

Case	α	β	TP	FN	FP	JSC	TP_{θ}	FN_{θ}	FP_{θ}	JSC_{θ}	RMSE	Min	Max	SD
1	0.000	0.000	0	5	0	0.000	0	1	0	0.000	-	-	-	-
2	1.000	1.000	5	0	0	1.000	1	0	0	1.000	0.000	0.000	0.000	0.000
3	0.364	0.364	5	0	0	1.000	1	0	0	1.000	3.317	1.414	4.123	0.935
4	0.308	0.308	4	1	1	0.667	1	0	0	1.000	3.240	1.414	4.123	1.018
5	0.052	0.052	3	2	2	0.429	1	0	0	1.000	3.109	1.414	4.123	1.121
6	0.256	0.256	8	2	2	0.667	2	0	0	1.000	2.894	1.414	4.123	0.822
7	0.026	0.026	3	7	2	0.250	1	1	0	0.500	3.109	1.414	4.123	1.121
8	0.052	0.026	3	2	7	0.250	1	0	1	0.500	3.109	1.414	4.123	1.121
9	0.168	0.140	6	4	4	0.429	2	0	1	0.667	2.887	1.414	4.123	0.828
10	0.142	0.089	3	7	7	0.176	1	1	1	0.333	2.646	2.236	2.828	0.279

REFERENCES

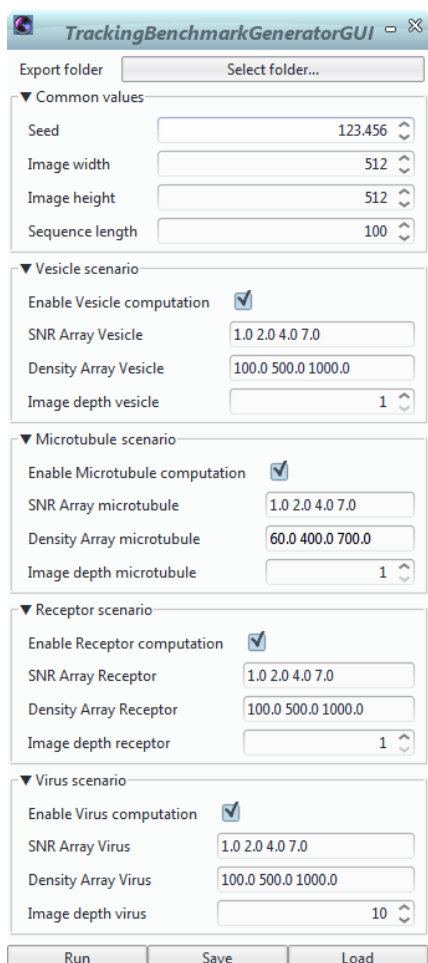
1. Fridling BE, Drummond OE. Performance evaluation methods for multiple-target-tracking algorithms. *Proceedings of SPIE: Signal and Data Processing of Small Targets* **1481**, 371–383 (1991).
2. Blackman S, Popoli R. *Design and Analysis of Modern Tracking Systems*. Artech House, Norwood, MA, USA (1999).
3. Rothrock RL, Drummond OE. Performance metrics for multiple-sensor, multiple-target tracking. *Proceedings of SPIE: Signal and Data Processing of Small Targets* **4048**, 521–531 (2000).
4. Meijering E, Smal I, Danuser G. Tracking in molecular bioimaging. *IEEE Signal Processing Magazine* **23**, 46–53 (2006).
5. Meijering E, Dzyubachyk O, Smal I, van Cappellen WA. Tracking in cell and developmental biology. *Seminars in Cell and Developmental Biology* **20**, 894–902 (2009).
6. Hoffman JR, Mahler RPS. Multitarget miss distance via optimal assignment. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* **34**, 327–336 (2004).
7. Schuhmacher D, Vo B-T, Vo B-N. A consistent metric for performance evaluation of multi-object filters. *IEEE Transactions on Signal Processing* **56**, 3447–3457 (2008).
8. Ristic B, Vo B-N, Clark D, Vo B-T. A metric for performance evaluation of multi-target tracking algorithms. *IEEE Transactions on Signal Processing* **59**, 3452–3457 (2011).
9. Born M, Wolf E. *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light*. 7th Edition. Cambridge University Press, Cambridge, UK (1999).
10. Ram S, Ward ES, Ober RJ. Beyond Rayleigh's criterion: A resolution measure with application to single-molecule microscopy. *Proceedings of the National Academy of Sciences of the United States of America* **103**, 4457–4462 (2006).
11. Munkres J. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics* **5**, 32–38 (1957).
12. Tan P-N, Steinbach M, Kumar V. *Introduction to Data Mining*. Addison Wesley, Boston, MA, USA (2005).

Supplementary Note 4: Simulation and Evaluation Software Tools

This document provides a description of the software developed for this study. Two software tools were developed. The first tool is able to generate the simulated image data and corresponding ground truth as described in **Supplementary Note 2**. The second tool computes the performance measures described in **Supplementary Note 3** for assessing the performance of a given particle tracking method with respect to the ground truth. Software implementations of the particle tracking methods themselves are available from the participating teams as indicated in **Supplementary Note 1**.

SOFTWARE FOR GENERATING SIMULATED IMAGE DATA

The software for generating simulated image data was written in the Java programming language as a plugin named **ISBI Challenge Tracking Benchmark Generator** for the open bioimage informatics platform Icy.¹ A snapshot of the plugin interface and a description of the parameters is given below.



The screenshot shows the 'TrackingBenchmarkGeneratorGUI' window. It features a top section for 'Export folder' with a 'Select folder...' button. Below this are four expandable sections: 'Common values', 'Vesicle scenario', 'Microtubule scenario', and 'Receptor scenario'. Each scenario section includes an 'Enable' checkbox, 'SNR Array' and 'Density Array' input fields, and an 'Image depth' spinner. The 'Virus scenario' is partially visible at the bottom. At the very bottom are 'Run', 'Save', and 'Load' buttons.

Parameter	Value
Export folder	Select folder...
Seed	123.456
Image width	512
Image height	512
Sequence length	100
Enable Vesicle computation	<input checked="" type="checkbox"/>
SNR Array Vesicle	1.0 2.0 4.0 7.0
Density Array Vesicle	100.0 500.0 1000.0
Image depth vesicle	1
Enable Microtubule computation	<input checked="" type="checkbox"/>
SNR Array microtubule	1.0 2.0 4.0 7.0
Density Array microtubule	60.0 400.0 700.0
Image depth microtubule	1
Enable Receptor computation	<input checked="" type="checkbox"/>
SNR Array Receptor	1.0 2.0 4.0 7.0
Density Array Receptor	100.0 500.0 1000.0
Image depth receptor	1
Enable Virus computation	<input checked="" type="checkbox"/>
SNR Array Virus	1.0 2.0 4.0 7.0
Density Array Virus	100.0 500.0 1000.0
Image depth virus	10

The interface is divided into several areas. At the top, the user can select an **Export folder** to store the simulated image data and the ground truth files.

The area **Common values** allows the user to set the values of the parameters used for all scenarios: The **Seed** to initialize the random processes driving the simulation (a different seed was used for the training data versus the competition data), the **Image width** (in pixels), the **Image height** (in pixels), and the **Sequence length** (in frames). Random number generation was done using the Flanagan library² implementing the appropriate algorithms for the different deviates needed.³

The next four areas specify the parameters specific to each of the scenarios (nicknamed Vesicles, Microtubules, Receptors, Viruses). It is possible to **Enable** the computation for each scenario separately and to specify the **SNR** levels (an array of values) and **Density** levels (an array of values) as well as the **Image depth** (number of slices).

The last area consists of buttons to **Run** the plugin and to **Save** or **Load** the parameter settings. Except for **Seed**, the values shown are the values used in this study.

For each image sequence (corresponding to a given scenario, SNR level, and density level) the plugin exports each slice and each frame as a separate file in the tagged image file format (TIFF). The ground truth is exported as an Extensible Markup Language (XML) file. This file specifies the spatial position (x, y, z) of each particle in each relevant frame (t). The spatial coordinates may be specified with floating-point precision and range from 0 (the center of the first pixel in any dimension) to N-1 (the center of the last pixel in any dimension). The temporal coordinate is an integer index, which may range from 0 (the first frame) to N-1 (the last frame), but only the relevant frames need to be specified. An example of an XML file is given below. More technical details are provided on the website.

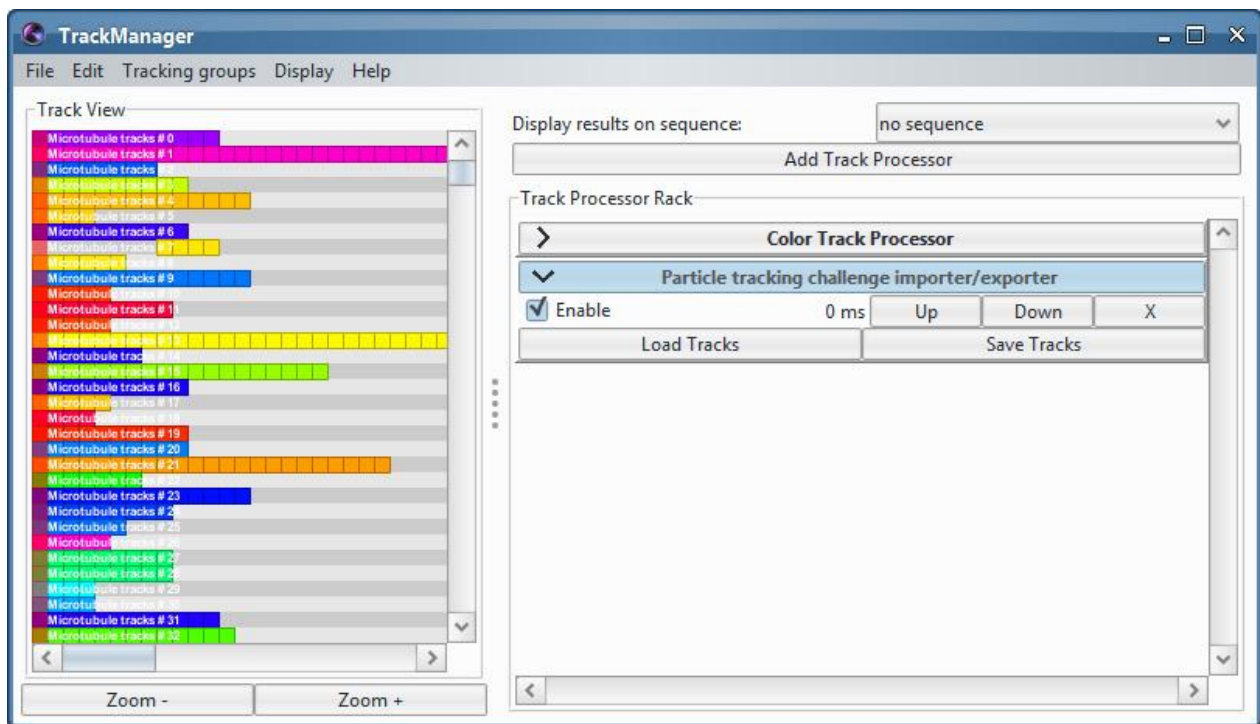
```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<root>
  <TrackContestISBI2012 snr="2" density="low" scenario="virus">
    <particle>
      <detection t="4" x="14" y="265" z="5.1"/>
      <detection t="5" x="14.156" y="266.5" z="4.9"/>
      <detection t="6" x="15.32" y="270.1" z="5.05"/>
    </particle>
    <particle>
      <detection t="14" x="210.14" y="12.5" z="1"/>
      <detection t="15" x="210.09" y="13.458" z="1.05"/>
      <detection t="16" x="210.19" y="14.159" z="1.122"/>
    </particle>
  </TrackContestISBI2012>
</root>
```

The particle tracks stored in the XML file can be visualized either in Icy, using the plugins **Track Manager**⁴ and **ISBI Challenge Tracks Importer**,⁵ or in ImageJ^{6,7} / Fiji⁸ using the MTrackJ plugin, after conversion to the MTrackJ Data Format (MDF) as described on the website.

SOFTWARE FOR COMPUTING PERFORMANCE MEASURES

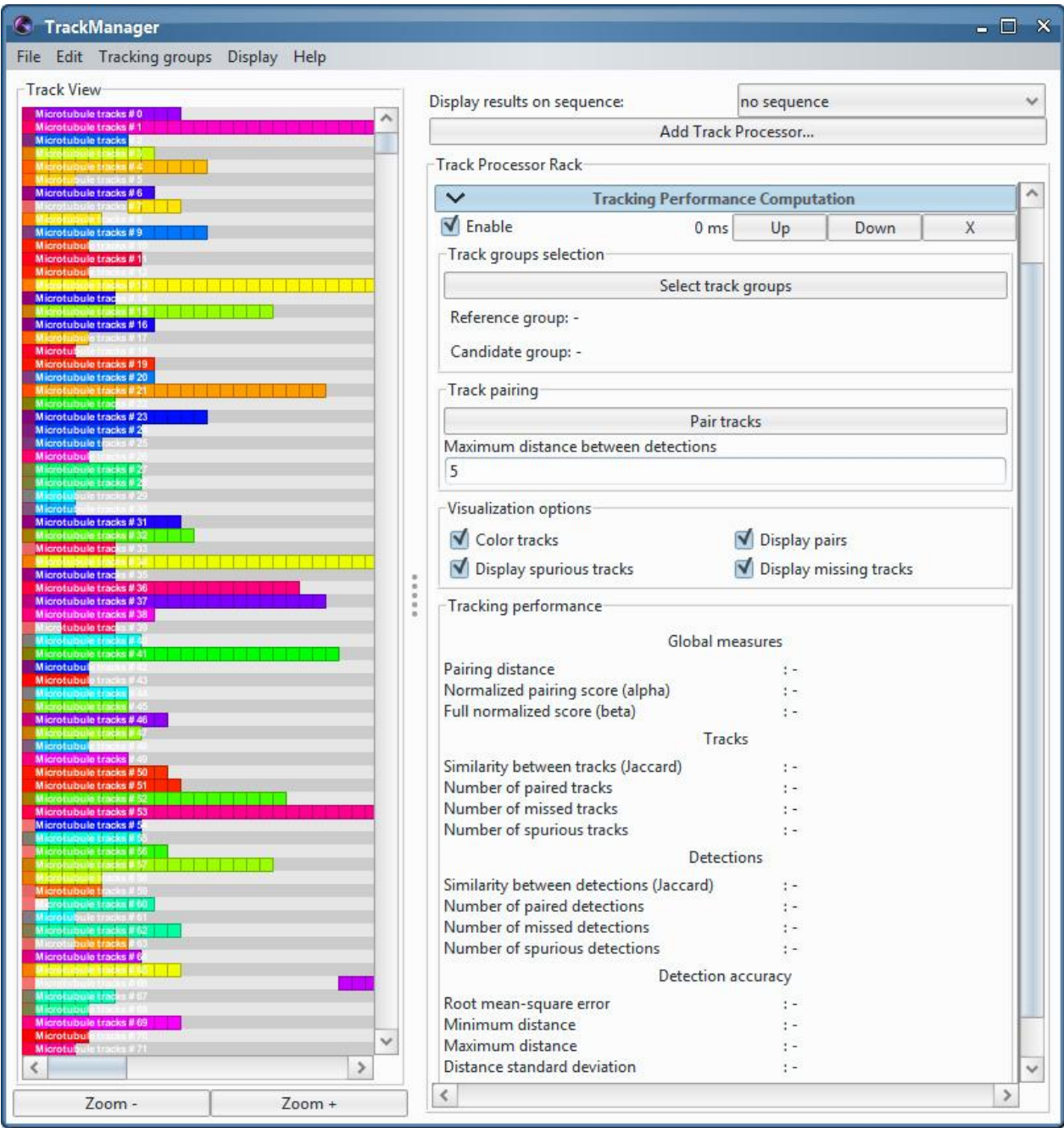
The software for computing the performance measures was written in the Java programming language, both as a stand-alone application, and as a plugin named **Tracking Performance Measures**⁹ for Icy. Both compute the full set of performance measures described in **Supplementary Note 3**. The stand-alone

application can simply be run from the command line as explained on the website. The Icy plugin requires the **Track Manager** plugin, which provides many useful tools for track visualization and track processing, and is already part of Icy. In order to load tracks into the **Track Manager** from an XML file that follows the format used in this study, the **ISBI Challenge Tracks Importer** plugin is needed, which is able to parse the file and convert the tracks to a data structure used by Icy. This importer can be selected in the **Track Manager** by clicking the **Add Track Processor** button, and the XML file can then be imported by clicking the **Load Tracks** button, shown in the snapshot below.



The **Track Manager** can load multiple track files, resulting in different track groups. In order to compute the performance of a given tracking method for a given image sequence, two files must be loaded: the XML file containing the output of the tracker, and the XML file containing the ground truth tracks. By clicking the **Add Track Processor** button of the **Track Manager** again, the mentioned plugin **Tracking Performance Measures** can be added, a snapshot of which is shown below. Clicking the **Select track groups** button of this plugin allows the user to select one of the track groups as the **Reference group** (the ground truth tracks) and the other as the **Candidate group** (the tracker output). The computation of the performance measures is started by clicking the **Pair tracks** button. The field **Maximum distance**

between detections specifies the gate ϵ described in the **Supplementary Note 3**. Upon completion, the resulting performance measures are listed in the area **Tracking performance**.



DOWNLOADING THE SOFTWARE

The software and detailed instructions are available from the website:

URL: <http://bioimageanalysis.org/track/>

Login: anonymous@bioimageanalysis.org

Password: erfg14d3

REFERENCES

1. De Chaumont F, Dallongeville S, Chenouard N, Hervé N, Pop S, Provoost T, Meas-Yedid V, Pankajakshan P, Lecomte T, Montagner YL, Lagache T, Dufour A, Olivo-Marin J-C. Icy: An open bioimage informatics platform for extended reproducible research. *Nature Methods* **9**, 690-696 (2012).
2. Flanagan Java Scientific Library. <http://www.ee.ucl.ac.uk/~mflanaga/java/>.
3. Press WH, Teukolsky SA, Vetterling WT, Flannery BP. *Numerical Recipes in C: The Art of Scientific Computing*. 2nd Edition. Cambridge University Press, Cambridge, UK, 1992.
4. Icy Plugin ID: N9W5B7. http://icy.bioimageanalysis.org/plugin/Track_Manager.
5. Icy Plugin ID: E5L1Q8. http://icy.bioimageanalysis.org/plugin/ISBI_Challenge_Tracks_Importer.
6. Abràmoff MD, Magalhães PJ, Ram SJ. Image processing with ImageJ. *Biophotonics International* **11**, 36-42 (2004).
7. Schneider CA, Rasband WS, Eliceiri KW. NIH Image to ImageJ: 25 years of image analysis. *Nature Methods* **9**, 671-675 (2012).
8. Schindelin J, Arganda-Carreras I, Frise E, Kaynig V, Longair M, Pietzsch T, Preibisch S, Rueden C, Saalfeld S, Schmid B, Tinevez J-Y, White DJ, Hartenstein V, Eliceiri K, Tomancak P, Cardona A. Fiji: An open-source platform for biological-image analysis. *Nature Methods* **9**, 676-682 (2012).
9. Icy Plugin ID: A9R8S3. http://icy.bioimageanalysis.org/plugin/Tracking_Performance_Measures.